

1971

Practical Applications of Advanced Control Theory.

William Harold Pusch Jr

Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation

Pusch, William Harold Jr, "Practical Applications of Advanced Control Theory." (1971). *LSU Historical Dissertations and Theses*. 2163.
https://digitalcommons.lsu.edu/gradschool_disstheses/2163

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

72-17,801

PUSCH, Jr., William Harold, 1945-
PRACTICAL APPLICATIONS OF ADVANCED CONTROL
THEORY.

The Louisiana State University and Agricultural
and Mechanical College, Ph.D., 1971
Engineering, chemical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

PRACTICAL APPLICATIONS OF ADVANCED
CONTROL THEORY

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Chemical Engineering

by

William Harold Pusch, Jr.
B.S., Louisiana State University, 1967
M.S., Louisiana State University, 1969
December, 1971

ACKNOWLEDGMENTS

The author wishes to express his gratitude to Dr. Cecil L. Smith under whose guidance and counsel this research was carried out.

Sincere thanks are also extended to the faculty and staff of the Department of Chemical Engineering with special recognition to Dr. Armando Corripio and Mr. James Paulsen for their assistance.

The author is grateful for the use of the computer facilities provided by Project THEMIS contract no. F44620-68-C-0021 administered through the Department of Defense by the U.S. Air Force Office of Scientific Research.

The author expresses his thanks to Mrs. Frank Hartley and Miss Hazel LaCoste for the typing and correcting of this manuscript. Appreciation is extended to the Dr. Charles E. Coates Memorial Fund of the L.S.U. Foundation, donated by Mr. George H. Coates, for the financial aid toward the publication of this manuscript.

PLEASE NOTE:

Some pages may have
indistinct print.

Filmed as received.

University Microfilms, A Xerox Education Company

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	v
LIST OF TABLES	viii
ABSTRACT	ix
CHAPTER	
I	
INTRODUCTION	1
II	
APPROXIMATE OPTIMAL CONTROL USING SENSITIVITY COEFFICIENTS	6
Sensitivity Coefficients	6
Second Order Model	14
Chemical Reactor	17
Control Strategy	20
Summary	35
Nomenclature	36
Literature Cited	38
III	
TUNING CONTROLLERS USING OPTIMAL CONTROL THEORY	39
General Process Control Problem	39
The Optimal Control Formulation	42
Approximating the Dead Time	43

	Page
Verification of the Method	53
Sampled-Data System	60
Summary	67
Nomenclature	69
Literature Cited	70
IV MULTIVARIABLE PROCESS CONTROL	72
Multivariable Systems	72
Relative Process Gain	77
The Interaction Index	85
Dynamic Control	87
Multivariable Control of an Absorber-Stripper Process Steady State Case	91
Multivariable Control of a Simple Model - Dynamic Case	100
Summary	123
Nomenclature	125
Literature Cited	126
V CONCLUSIONS	127
APPENDIX	
A DEVELOPMENT OF CONTROL LAW USING KALMAN'S ALGORITHM	131
B SUM OF ROWS AND COLUMNS OF RELATIVE GAIN MATRIX EQUAL ONE	139
C USING CHESS ON THE SIGMA 5	145
D COMPUTER PROGRAMS	170
VITA	212

LIST OF FIGURES

Figure		Page
2.1	Physical configuration for method of inverse sensitivity	10
2.2	Chemical Reactor Process	18
2.3a	Input changes in set point on reactor temperature	22
2.3b	Response of manipulated variable	23
2.3c	Reactor Temperature Response to Optimal Switching Control	24
2.4a	Variation of Model Parameter A	26
2.4b	Variation of Model Parameter B	27
2.4c	Variation of Model Parameter K	28
2.5	Typical sampled data control system	33
2.6	Temperature response for set point change using parameters from sensitivity analysis and three different control strategies	34
3.1	General process control problem	41
3.2	Alternate representation of the disturbance	41
3.3	Dimensionless responses to a step change in load for various values of p'	49
3.4a	Optimum Control parameters for various values of p'	50
3.4b	Values for Reset Time	51
3.4c	Values for Rate Time	52
3.5	Chemical reactor	54

Figure		Page
3.6a	Open loop response to step change in the water flow rate	55
3.6b	Open loop response to change in flow rate of reactants	56
3.7a	Response for setpoint change with continuous feedback PID controller	58
3.7b	Responses for load change with both PID and PCID continuous controllers	59
3.8	Response for set point change with continuous PCID controller	61
3.9a	Sampled-data system	62
3.9b	Continuous approximation	62
3.9	Dead time approximation to sample and hold	62
3.10a	Response for setpoint change with PID controller	64
3.10b	Response to setpoint change with PCID controller	65
3.10c	Responses to load change for both PID and PCID controller	66
4.1	Typical multivariable system	74
4.2	Closed loop control of system with two inputs/two outputs	75
4.3	Diagram for Blending System	80
4.4	Absorber-stripper natural gas process	93
4.5	Model diagram for multivariable study	104
4.6	Set point response for closed loop on Y1-F	106

Figure		Page
4.7	Set point responses for closed loop on Y2-T	107
4.8	Set point response for both loops closed	108
4.9	Diagram of system with steady state decoupler	109
4.10	Response to step input on M_1 with steady state decoupler	111
4.11	Response to step input on M_2 with steady state decoupler	112
4.12	Dynamic decoupler	114
4.13	Response to step input on M_1 with dynamic decoupler	115
4.14	Response to step input on M_2 with dynamic decoupler	116
4.15	Set point response for closed loop on Y1-F with dynamic decoupler	118
4.16	Set point response for closed loop on Y2-T with dynamic decoupler	119
4.17	Response to set point change on Y1 with both loops closed and dynamic decoupler	120
4.18	Response to set point change on Y2 with both loops closed and dynamic decoupler	121
4.19	Response to change in set point on Y1 and Y2 with dynamic decoupler	122
A2.1	Sketch of Ideal Response for Kalman Algorithm	133
A3.1	Overlay Structure for CHESS on the Sigma 5	169

LIST OF TABLES

Tables		Page
II-1	Sensitivity Coefficients and Pattern Search to Determine Correct Model Parameters for Second Order System	15
IV-1	Stream Conditions for Absorber-Stripper Process	94
IV-2	Steady State Concentrations of the Controlled Variables	96
IV-3	Model Parameters for Multivariable Study	103

ABSTRACT

This work is written in an effort to illustrate some of the problems involved in the design of control systems for chemical processes and to present approaches that can be taken to overcome these problems. Adaptive control, nonlinear control, optimal control, direct digital control, and multivariable control are discussed with the emphasis on applying them to typical chemical processes.

An approach to the identification of the process is presented first. The method of inverse sensitivity is applied to a chemical reactor process, determining the parameters for a general second order model. With the sensitivity coefficients continuously updating or adapting the model, it was found that optimal control or other advanced control strategies could be applied to the process to produce near optimal responses. Specifically, the three control strategies implemented were the conventional PI controller, the Kalman algorithm which is a sampled-data technique, and the optimal switching times as determined from optimal control theory. The responses for the chemical reactor due to set point changes were compared for each of these control methods.

Extending the control aspects of the work, tuning parameters were then determined by applying optimal control theory. The

controller settings for a conventional PID analog controller are obtained from a first order lag plus dead time model and an optimal control approach. The model equations were transformed into a state space configuration and the coefficients for the feedback control law determined from the linear regulator problem. The controller constants were found for values of the model gain, time constant, and dead time and also for the control weighting factor p in the cost function. The controller constants were tested on a chemical reactor process and proved to yield desirable responses for both load and set point changes. A similar algorithm which uses the feedback variable instead of the error signal was also implemented along with a sampled-data approach. For the reactor system, the responses of those two strategies were compared to those of the PID controller.

To be applicable to chemical processes, the multivariable nature of the process must be considered in designing the control system. The proper pairing of controlled and manipulated variables is discussed and determined by means of the relative process gain matrix. In order to investigate the variable pairing of an absorber-stripper process, CHESS, a chemical process simulation program, is utilized to find the steady state gain matrix. From this matrix the relative gain matrix is calculated and the effects of variable pairings observed. Finally, the dynamic case is investigated by means of a theoretical model. Advanced control concepts such as a steady state decoupler and tuning compensation for interacting loops were found to produce unsatisfactory responses. Only

the dynamic decoupler compensated for variable interaction and enabled the tuning of controllers for multivariable processes by tuning each loop separately.

Chapter I

INTRODUCTION

Advances in control theory have progressed significantly during the last three decades. These advances have been brought about primarily by the growing interest in aircraft, missile and spacecraft guidance systems. Also, the advent of the high speed digital computer has made possible the complex or lengthy calculations and the storage of large data banks necessary for the solution of control problems. Thus, it was natural for those in the chemical process industries to try to apply these concepts developed in the field of control to their systems in order to maximize their profits or to minimize their expenditures. However, several drawbacks or limitations prohibited the direct application of advanced control techniques to the process industries. First, most of the process systems are large, bulky and slow in reacting to inputs. This is caused by their large capacities and throughputs which possess large dead times and time constants. Another reason for the lag in control applications is that most chemical processes are quite complex and very difficult to model. This inability to predict or describe the behavior of the process is in fact, the most

significant restriction of the application of modern control techniques. The result is that approximations must be made and special techniques must be applied in order to transform a process into a system in which these advanced concepts can be applied. This will be the goal of this thesis: to present methods which will allow some of the new developments in the field of control to be applied to processes in the chemical industries.

Some of the recent achievements in control strategies have been made in adaptive control, nonlinear control, optimum control, direct digital control, and multivariable control. All of the above will be discussed in the following chapters along with test cases in which some form of each will be used to control a typical chemical process. The emphasis will be on the application of the particular technique in order to design a "controllable" system.

Chapter II deals with the problem of identification; that is, fitting a particular model to a process. In the case discussed sensitivity coefficients are used to determine the parameters for a general second order model. The techniques update the model at specified intervals, insuring that the model has current parameters at new operating levels. Therefore, any control scheme based on this process model will become, by definition, an adaptive controller. From the model parameters,

three control strategies were implemented in order to determine whether the adaptive system would yield satisfactory control. A conventional PI controller which was tuned optimally was applied to the test case and the responses were used as a basis for comparison. An optimal control scheme determined the switching times from a minimum time formulation and the model parameters. Finally, a control algorithm presented by Kalman and designed to drive the system to a new setpoint in two sampling times was incorporated into the system. The algorithm had to be modified somewhat due to the limits imposed on the manipulated variable. From the responses resulting from these three control configurations, the justification for this type of model reference adaptive control strategy will be determined.

With the availability of a scheme by which model parameters can be updated, a natural question to ask is why can't optimal control theory be used to tune a conventional control algorithm as opposed to the three approaches used in Chapter II to design the algorithm. In Chapter III optimal control theory is employed to find tuning relationships for a conventional PID analog controller. Starting with a first order lag plus dead time model, the dead time term is eliminated by means of a Pade' approximation. The model is then written as a linear state equation and through a formulation proposed by Johnson, state regulator theory will yield settings for a PID controller. The

approach should result in a tuning technique based on a performance specification that contains a parameter that the user can vary to "tighten up" or "loosen" the controller settings. This chapter considers both set point and load control problems and extends the results to discrete systems.

In Chapters II and III of this dissertation, the control loops have been treated as if each were "stand-alone", i.e., not influenced by control actions taken in other loops. Chemical processes rarely exhibit this characteristic. Whenever two or more control loops are placed on a single unit, the loops are coupled through the process itself. To be able to successfully utilize the techniques presented in the previous chapters, as well as in most articles that have appeared in the literature, an approach must be used by which this interaction as well as the individual loop characteristics can be considered in designing the overall control system.

Chapter IV deals with the problem of multivariable control. The first problem considered is the proper pairing of controlled and manipulated variables. This is accomplished through the use of the relative process gain matrix which is calculated from the steady state open loop gains. This chapter explains how a large simulation program can be useful in these initial stages of controller design. CHESS, a Chemical Engineering Process Simulator is used to illustrate the technique for an

absorber-stripper process. Some dynamic considerations for multivariable control are investigated by means of a general interacting model consisting of two inputs and two outputs. This model is first controlled with a conventional PI controller. An effort is made to reduce the system to two non-interacting loops by inserting into the configuration first, a steady state decoupler, then, a dynamic decoupler. Process responses for these systems are then compared and the effects of each are discussed. Finally, advanced tuning techniques for interacting loops are applied to the test model.

Chapter II

APPROXIMATE OPTIMAL CONTROL USING SENSITIVITY COEFFICIENTS

The application of optimal control theory to actual chemical processes have been based on approximate, low order, linear mathematical models rather than the true process model, which is frequently unknown. The model must be sophisticated enough to fit the process well, yet simple enough so that its parameters can be evaluated without an excessive computational effort. This chapter describes a model reference adaptive control strategy using sensitivity coefficients to update the model and optimal control theory to determine the control strategy, which is applied to a stirred chemical reactor.

The discussion will begin with a review of sensitivity coefficients.

Sensitivity Coefficients

Sensitivity or influence coefficients are defined as (1,2,3)

$$\mu_{q_i} = \partial c / \partial q_i \quad (2.1)$$

where μ_{q_i} is the sensitivity coefficient
 c is the system output
 q_i is the parameter of the system

The general equation for a second order system is

$$\ddot{c} + A\dot{c} + Bc = K f(t) \quad (2.2)$$

where A, B, and K are non-interacting system parameters and \dot{c} and \ddot{c} are the first and second derivatives of the system output.

If the partial derivatives with respect to A, B, and K are taken, the following system of equations is obtained:

$$\frac{\partial}{\partial A} \left(\frac{\partial^2 c}{\partial t^2} \right) + \frac{\partial}{\partial A} \left(A \frac{\partial c}{\partial t} \right) + \frac{\partial}{\partial A} (B c) = \frac{\partial}{\partial A} (K f(t)) \quad (2.3a)$$

$$\frac{\partial}{\partial B} \left(\frac{\partial^2 c}{\partial t^2} \right) + \frac{\partial}{\partial B} \left(A \frac{\partial c}{\partial t} \right) + \frac{\partial}{\partial B} (B c) = \frac{\partial}{\partial B} (K f(t)) \quad (2.3b)$$

$$\frac{\partial}{\partial K} \left(\frac{\partial^2 c}{\partial t^2} \right) + \frac{\partial}{\partial K} \left(A \frac{\partial c}{\partial t} \right) + \frac{\partial}{\partial K} (B c) = \frac{\partial}{\partial K} (K f(t)) \quad (2.3c)$$

Since only c is a function of the three parameters, the order of differentiation can be changed and the equations can be simplified to:

$$\frac{\partial^2}{\partial t^2} \left(\frac{\partial c}{\partial A} \right) + A \frac{\partial}{\partial t} \left(\frac{\partial c}{\partial A} \right) + \frac{\partial c}{\partial t} + B \left(\frac{\partial c}{\partial A} \right) = 0 \quad (2.4a)$$

$$\frac{\partial^2}{\partial t^2} \left(\frac{\partial c}{\partial B} \right) + A \frac{\partial}{\partial t} \left(\frac{\partial c}{\partial B} \right) + B \left(\frac{\partial c}{\partial B} \right) + c = 0 \quad (2.4b)$$

$$\frac{\partial^2}{\partial t^2} \left(\frac{\partial c}{\partial K} \right) + A \frac{\partial}{\partial t} \left(\frac{\partial c}{\partial K} \right) + B \left(\frac{\partial c}{\partial K} \right) = f(t) \quad (2.4c)$$

The three sensitivity coefficients are defined as:

$$\mu_A = \partial c / \partial A \quad (2.5a)$$

$$\mu_B = \partial c / \partial B \quad (2.5b)$$

$$\mu_K = \partial c / \partial K \quad (2.5c)$$

Thus the following three differential equations are obtained:

$$\ddot{\mu}_A + A\dot{\mu}_A + \dot{c} + B\mu_A = 0 \quad (2.6a)$$

$$\ddot{\mu}_B + A\dot{\mu}_B + B\mu_B + c = 0 \quad (2.6b)$$

$$\ddot{\mu}_K + A\dot{\mu}_K + B\mu_K = f(t) \quad (2.6c)$$

The initial conditions for these second-order differential equations must be

$$\mu_i(0) = 0 \quad (2.7a)$$

$$\dot{\mu}_i(0) = 0 \quad (2.7b)$$

because at $t = 0$ there is no information available about the effect of the system output due to parameter changes. Thus, only

after the system responds to a system input can sensitivity information be generated. Notice from the sensitivity equations that if the system output, its derivative, and the forcing function are zero, no coefficients can be obtained. In other words, a forcing function must be present to obtain the effect of parameter K and a system that is not at equilibrium ($\dot{c} \neq 0$) is needed to determine the effects of A and B . These two conditions are apparent from the definitions of the sensitivity coefficients.

The three auxiliary or slave equations (2.6) can be solved simultaneously with the model equation (2.2) to obtain the three sensitivity coefficients. A number of methods have been proposed using these coefficients for parameter identification.

Watson (8) in his paper on sensitivity functions in adaptive control systems uses the coefficients in an optimal control formulation. He adjoins the sensitivity term to his cost function and minimizes at $t = t_f$, resulting in a fixed terminal point problem. Watson then minimizes the criterion function with an optimization program (gradient acceleration technique) determining the control law. He also treats the identification problem with sensitivity coefficients "adapting" the system model but, in this problem too, an optimization technique is used to minimize the cost function directly.

Tomovic (7) proposes a method of inverse sensitivity which is used to solve the system in Figure 2.1.

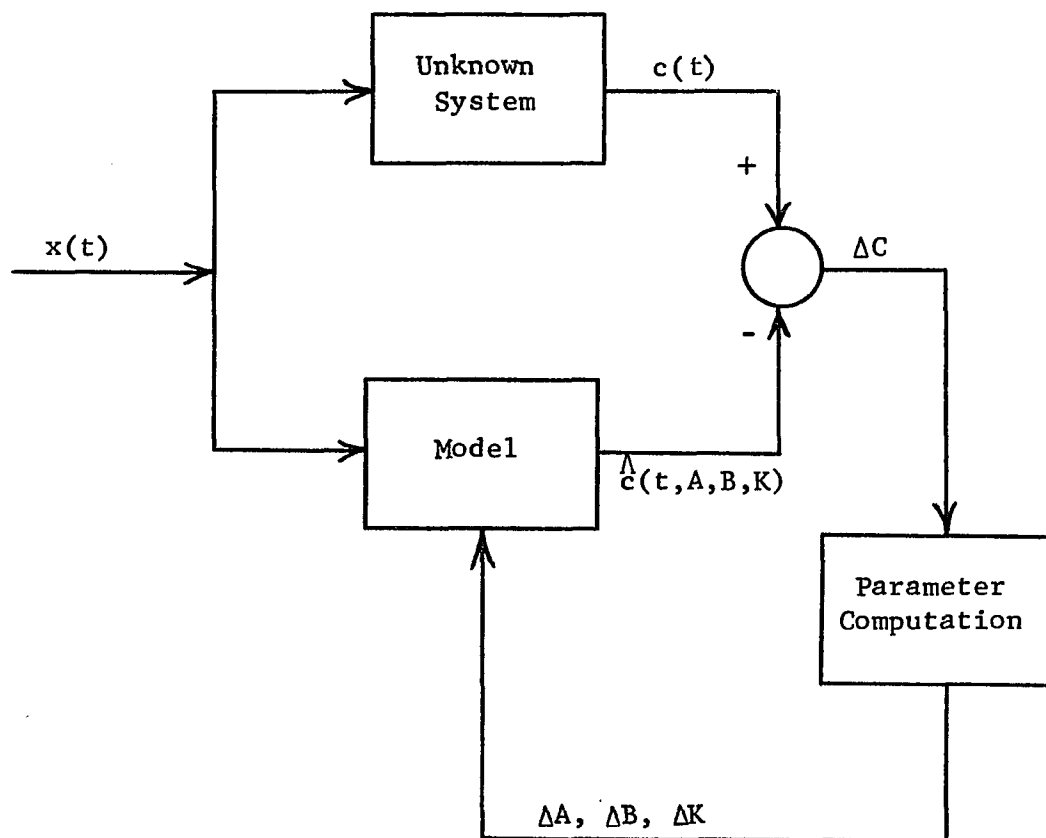


Figure 2.1. Physical configuration for method of inverse sensitivity.

A commonly used criterion or cost function is the integral of the square of the error (ISE) which is given by

$$\int_0^T \Delta C^2 dt \quad (2.8)$$

where ΔC is the difference between the system and model outputs, that is, $c(t)$ and $\Delta c(t, A, B, K)$. Often in optimization a constraint is added or adjoined directly to the primary cost function. In the problem considered here it is desired that the parameter changes ΔA , ΔB , and ΔK will become zero. Therefore, from a linear expansion of the perturbation of ΔC , the change in output due to changes in the system parameters may be defined as ΔC_q and given

$$\Delta C_q = \sum_{i=1}^N \left[\partial c / \partial q_i \right] \Delta q_i \quad (2.9)$$

where N equals the number of parameters to be determined. The new cost function to be minimized is then given by

$$\int_0^t \left[\Delta C + \Delta C_q \right]^2 dt \quad (2.10)$$

It is seen that $\partial c / \partial q_i$ are just the sensitivity coefficients and Δq_i are the changes of the system parameters. Since the criterion function is to be minimized, its derivative is taken with respect to the parameter changes to be determined and the

result set equal to zero. For three parameters, three algebraic equations are obtained with three unknowns and can be solved for the parameter increments that would minimize the cost function.

The resulting equations are:

$$\int_0^T [\mu_A \mu_A \Delta A + \mu_A \mu_B \Delta B + \mu_A \mu_K \Delta K + \mu_A \Delta c] dt = 0 \quad (2.11a)$$

$$\int_0^T [\mu_B \mu_A \Delta A + \mu_B \mu_B \Delta B + \mu_B \mu_K \Delta K + \mu_B \Delta c] dt = 0 \quad (2.11b)$$

$$\int_0^T [\mu_K \mu_A \Delta A + \mu_K \mu_B \Delta B + \mu_K \mu_K \Delta K + \mu_K \Delta c] dt = 0 \quad (2.11c)$$

The parameters are assumed to be constant, allowing ΔA , ΔB , and ΔK to be removed from the integrations. The equations can be solved to obtain the change that should be made for each parameter. The two assumptions, that of a linear approximation and also that of constant parameters, limit the solution to regions within a reasonable distance from the actual parameter values. At large deviations convergence is not assured and instability may result.

It will be shown that for small parameter deviations Tomovic's linear approach of solving the set of algebraic equations will converge rapidly, usually in one iteration. For larger parameter changes which are outside of the region of convergence, the first order linear approximation must be converted into a steepest-descent method. In this way convergence is assured even though slow convergence results. The steepest descent method

used will be the one proposed by Marquardt (3) because it is really a hybrid of the two methods.

The linear method requires the expansion of the criterion function in a Taylor's series. The derivatives of this expression are then taken with respect to the unknown parameters and these derivatives are set equal to zero. The result, known as the Gauss method, requires the solving of these equations for the unknown parameters.

$$\underline{A} (\Delta \underline{Q}_i) = \underline{B} \quad (2.12)$$

where \underline{A} and \underline{B} are matrices corresponding to the integrations of the sensitivity coefficients in this problem. However, in the gradient methods, the derivatives are not set equal to zero and steps are taken in the direction of the negative gradient of the cost function. Marquardt suggests that a combination of the two methods be used. At distances outside the normal region of convergence (large parameter changes) a gradient approach is applied but as the solution enters the neighborhood of the optimal parameters, the Gauss method is used to speed convergence. This is accomplished by adding a constant factor λ which operates as a Lagrangian multiplier. The algorithm is written as

$$(\underline{A} + \lambda \underline{I}) \Delta \underline{Q}_i = \underline{B} \quad (2.13)$$

As the λ values increase a gradient method is obtained. When λ is zero, the algorithm reduces to the Gauss method, Equation (2.12). Although Marquardt presents a method for determining how to choose the optimum λ values, good results were obtained with a simple trial and error approach.

Second Order Model

The technique using the sensitivity coefficients was tested with two different approaches. First, a second order equation with known parameters was used as the unknown process and an equation of the same form was taken as the model. Starting with assumed values of the parameters, the method outlined above was successively applied to the same process response. If the parameters, A, B, and K in the model are found to converge to the known values in the process, the method will be considered applicable. The results (see Table 2.1) show that the approach was successful in determining the correct parameters. However, the solution was iterative and the number of iterations depended on the initial guesses of the model parameters. If these starting points were within a reasonable distance from the actual values, the solution converged rapidly. At other starting points the solution seemed to oscillate and at still others the method completely broke down. This was due to the method trying to over-compensate with parameter changes larger than necessary, and was overcome by using Marquardt's method as described previously.

Results - Table II-1

Sensitivity Coefficients and Pattern Search to
Determine Correct Model Parameters for 2nd Order System

Correct model A = .7
 parameters

B = .05

K = -.08

<u>Starting Values</u>			<u>Sensitivity Coefficients</u> <u># of Iterations</u>	<u>Pattern Search</u> <u># of Iterations</u>
<u>K</u>	<u>A</u>	<u>B</u>		
-.001	.01	.1	15	3371
-.001	1.	.1	13	2895
-.001	.01	.001	25 ¹	1905
-.001	1.	.001	35 ¹	4360
-.1	.01	.1	9	957
-.1	.01	.001	12	277 ²
-.1	1.	.001	8	287 ²
-.1	1.	.1	5	902

¹ The sensitivity coefficient approach require λ value of 2.
 added to diagonal elements of A matrix in order to maintain stability.

² Pattern search did not obtain correct model parameters.

Table (2.1) compares the number of iterations to the results of Pattern Search (8) with the same starting points. Notice that for two sets of starting points the pattern search converges to incorrect parameters whereas the sensitivity coefficients find the correct values.

If a model with a different form is used, the sensitivity method can still be applied. For example, the second-order model can be written in terms of two time constants

$$\frac{c}{r} = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2.14)$$

or in terms of gain, natural frequency, and damping ratio

$$\frac{c}{r} = \frac{K\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \quad (2.15)$$

Even though one can convert from one set of these parameters to another, the sensitivity coefficients seem to work best on the simplest method in which the parameters are non-interacting. For this reason, the original model with the parameters A, B, and K was used.

If the starting values of the parameters are close to the actual values, only a once-through identification using the sensitivity coefficients is necessary. This makes it unnecessary

to store the process response, thus saving core storage. This condition should hold for successive identifications of a changing process system, but may not hold for the initial identification unless good estimates of the parameters are available.

Chemical Reactor

A second method of testing the sensitivity coefficients was that of trying to fit a second order model to a non-linear process and noting if the parameters converge to acceptable values. For this problem the stirred tank reactor (see Figure 2) was considered. In the system a second order chemical reaction takes place

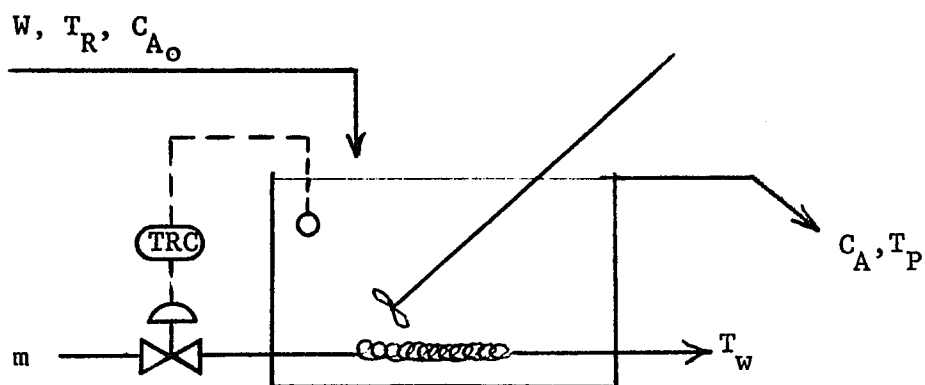


The reaction is exothermic with heat of reaction ΔH equal to -12,000 Btu/lb. mole. There is no volume change with reaction. The rate expression is $r_a = -kC_A^2$ with r_a the rate of disappearance of A. The rate constant k follows the Arrhenius expression in temperature dependence

$$k = k_c e^{-a/T}$$

$$k_c = 8.33 \times 10^8 \text{ cu ft/lb.mole min.} \quad (2.17)$$

$$a = 1400^\circ\text{R}$$



System Parameters

$V_t = 13.38 \text{ ft}^3$
 $V_B = 8.64 \text{ ft}^3$
 $A_B = 200.0 \text{ ft}^2$
 $\rho_t = 55 \text{ lb/ft}^3$
 $U = 75.0 \text{ BTU/hr}^\circ\text{F f}^2$
 $C_p = .9 \text{ BTU/lb}^\circ\text{F}$
 $c_p = 1.0 \text{ BTU/lb}^\circ\text{F}$
 $\rho_j = 62.4 \text{ lb/ft}^3$
 $\Delta H = -12,000 \text{ BTU/lb-mole}$

Steady State Conditions

$C_{A_0} = 0.5975 \text{ lb-mole/ft}^3$
 $C_A = 0.2068 \text{ lb-mole/ft}^3$
 $T_{w_i} = 80^\circ\text{F}$
 $T_w = 218.86^\circ\text{F}$
 $T_R = 175^\circ\text{F}$
 $T_P = 205.31^\circ\text{F}$
 $m = 26.89 \text{ lb/min}$
 $w = 73.5 \text{ lb/min}$

Figure 2.2. Chemical Reactor Process.

Assumptions are that the temperature of the tube bundle is uniform, the reactor is well mixed and the heat loss to the surroundings is negligible. Three equations are needed to describe the system. An unsteady state enthalpy balance on the jacket yields

$$wC_p(T_R - T_p) + (-\Delta H)kC_a^2 V_t - UA_B(T_p - T_w) = V_t \rho_t C_p (dT_p/dt) \quad (2.18)$$

An unsteady state balance on component A in the kettle indicates

$$w(C_{a_o} - C_a)/\rho_t - V_t kC_a^2 = V_t (dC_a/dt) \quad (2.19)$$

For the tube bundle an unsteady state enthalpy balance gives

$$mC_p(T_{w_i} - T_w) + UA_B(T_p - T_w) = V_B \rho_B C_p (dT_w/dt) \quad (2.20)$$

These three equations will describe the system to be investigated. The same second order model that was used in the preceeding case was applied to the process, and the solution was generated using a fourth-order Runge-Kutta numerical technique. Difficulty arose in that starting values had to be approximated for convergence and when these values were not sufficiently close to those of the solution, instabilities resulted. For these cases, Marquardt's method had to be used.

Control Strategy

To evaluate the utility of the sensitivity coefficients in the design of controllers, three different approaches were used to design control laws for the chemical reactor in Figure 2.2. The reactor temperature T_p is controlled using the jacket water flow rate m as the manipulated variable. In all cases, the manipulated variable was constrained between 0 and 100 lbs/min. In a paper by Latour (2), a time-optimal switching curve is used to find the control strategy for a set-point change. If a set-point change is desired for a second order process, optimal control theory for a time optimal system dictates that the manipulated variable be operated bang-bang. The two switching times for the bang-bang controller are determined from the two time constants and system gain in the process model. When these three parameters fit the process exactly, the time-optimal response for the system should result. In Latour's presentation a non-linear statistical regression program was used to fit the model parameters to the response of the process. It is believed that the sensitivity approach might obtain the parameters more quickly and also that it could be adapted to perform the analysis in the on-line control scheme. The method was applied to the equations for the backmix reactor and the second order model parameters A , B , and K were found. These were then converted to the form of two time constants and a gain and the optimal switching times were determined from the following equations:

$$\left[\frac{K-k-(r_o/K_p-k) \exp(-t_2/bT)}{K-r/K_p} \right]^b = \frac{K-k-(r_o/K_p-k) \exp(-t_2/T)}{K-r/K_p} \quad (2.21)$$

where K and k are the upper and lower limits on the manipulated variable

r_o is the value of the controlled variable at steady-state

r is the value of the new setpoint

K_p is the process gain

T is the major time constant

bT is the minor time constant

The value of the first switch t_2 is found by solving this implicit equation using the method of halving-the-increment. The value of the second switch t_4 is found using the equation

$$t_4 = T \left(\ln \left[\frac{r_o/K-k-(K-k) \exp(t_2/T)}{r/K_p-K} \right] \right) \quad (2.22)$$

Both of these equations assume that r is less than r_o . If $r > r_o$ interchanging K and k in the above equations would yield the correct results.

Figure 2.3 illustrates the effectiveness of this control strategy for the chemical reactor described above. The series of step inputs shown in Figure 2.3a was applied to the process.

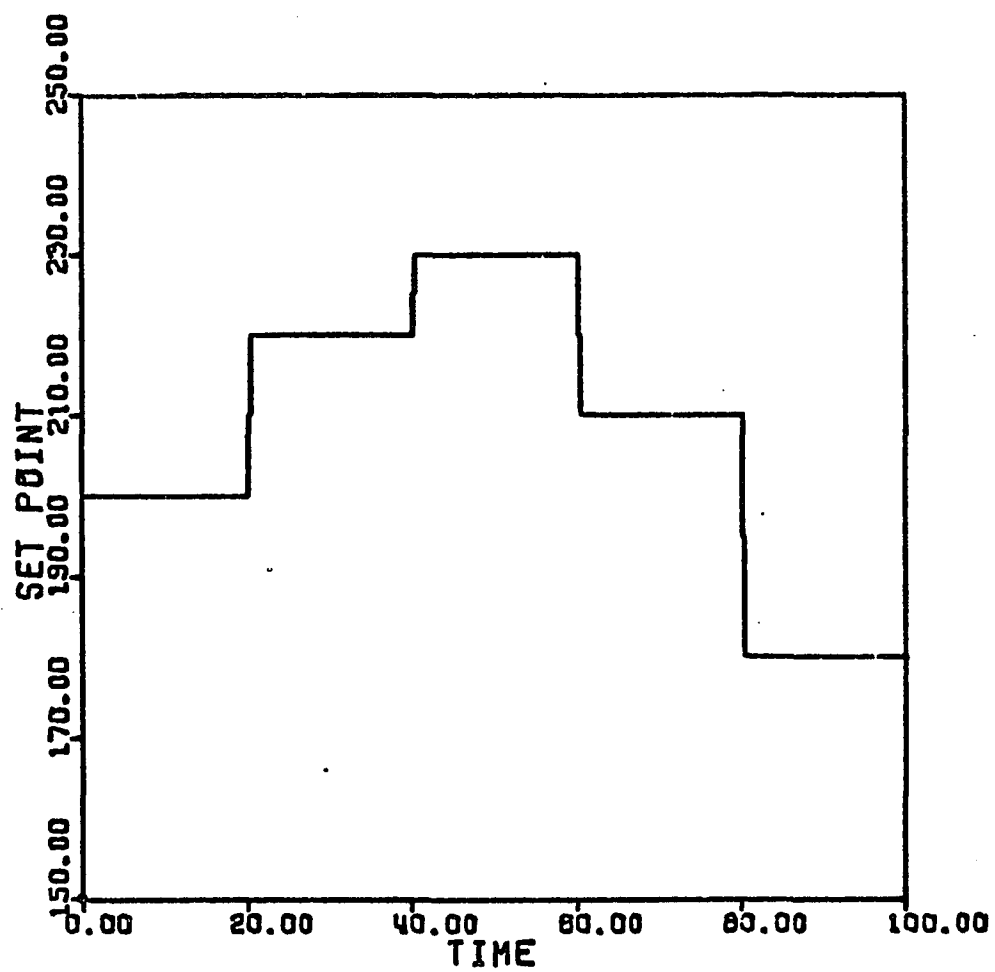


Figure 2.3a. Input changes in set point on reactor temperature.

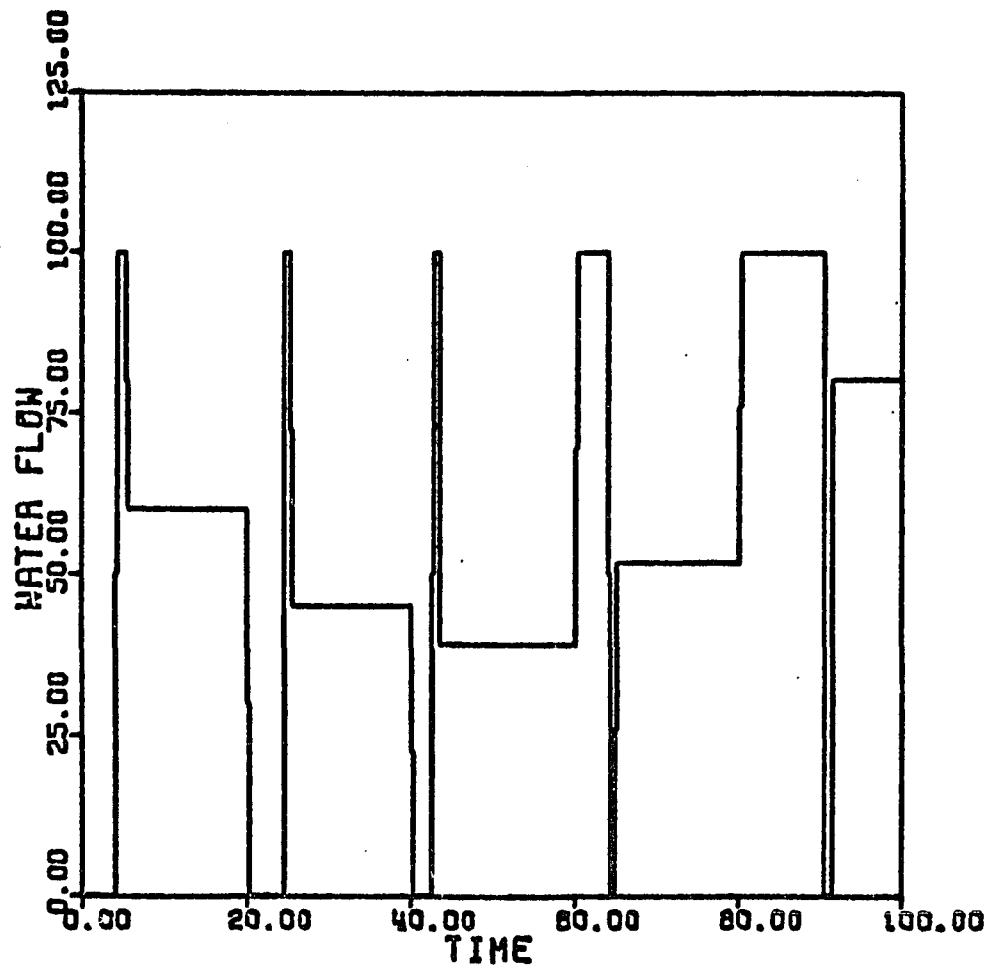


Figure 2.3b. Response of manipulated variable.

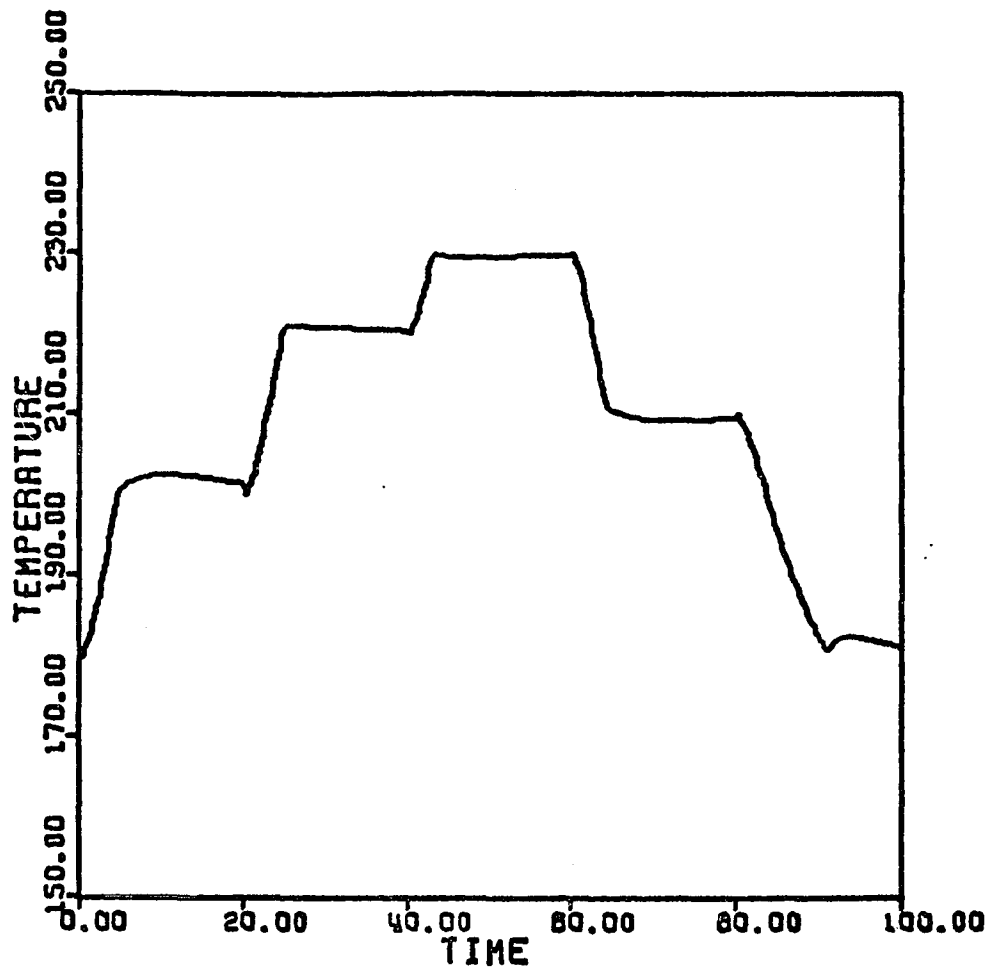


Figure 2.3c. Reactor Temperature Response to Optimal Switching Control

Figure 2.3b shows the input to the process and Figure 2.3c shows the resulting response. Just prior to each change in set-point, the coefficients A, B, and K in the model were re-evaluated. The results are shown in Figure 2.4. For the first step input the control was calculated using estimates known to be close to their true values at the level of operation, but were not exact.

In the optimal-time formulation the maximum and minimum values of the manipulated variable are fixed and the two switching times are determined for any set point changes. Kalman (1) considers the converse of this problem, where the two switching times are fixed and the maximum and minimum values of the control variable are obtained from a feedback algorithm.

The algorithm for the Kalman controller is derived from the physical configuration seen in Figure 2.5. The plant or process transfer function which will be replaced with the derived model equation is given by equation 2.2 and can be written as

$$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2.23)$$

Using a zero-order hold, the z-transform expression for the plant is

$$HG(z) = z \left[\frac{1 - e^{-Ts}}{s} \right] G(s) = \frac{P(z)}{Q(z)} \quad (2.24)$$

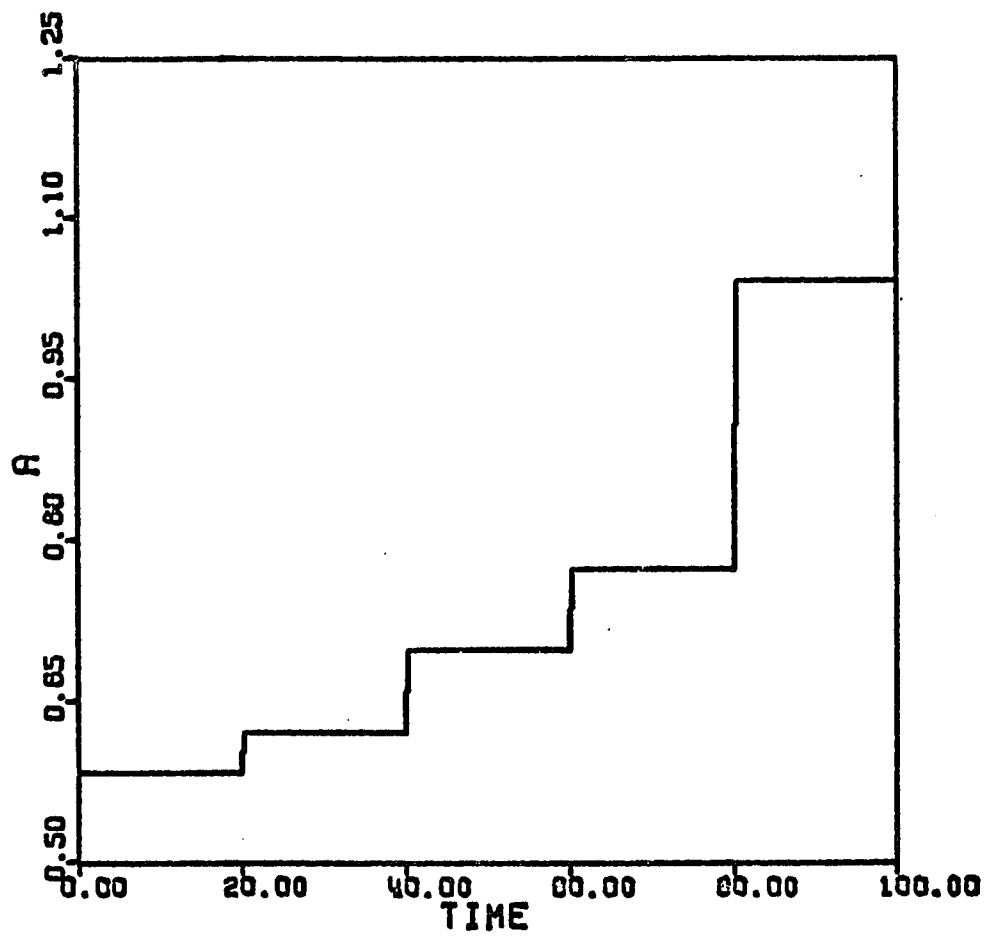


Figure 2.4a. Variation in Model Parameter A.

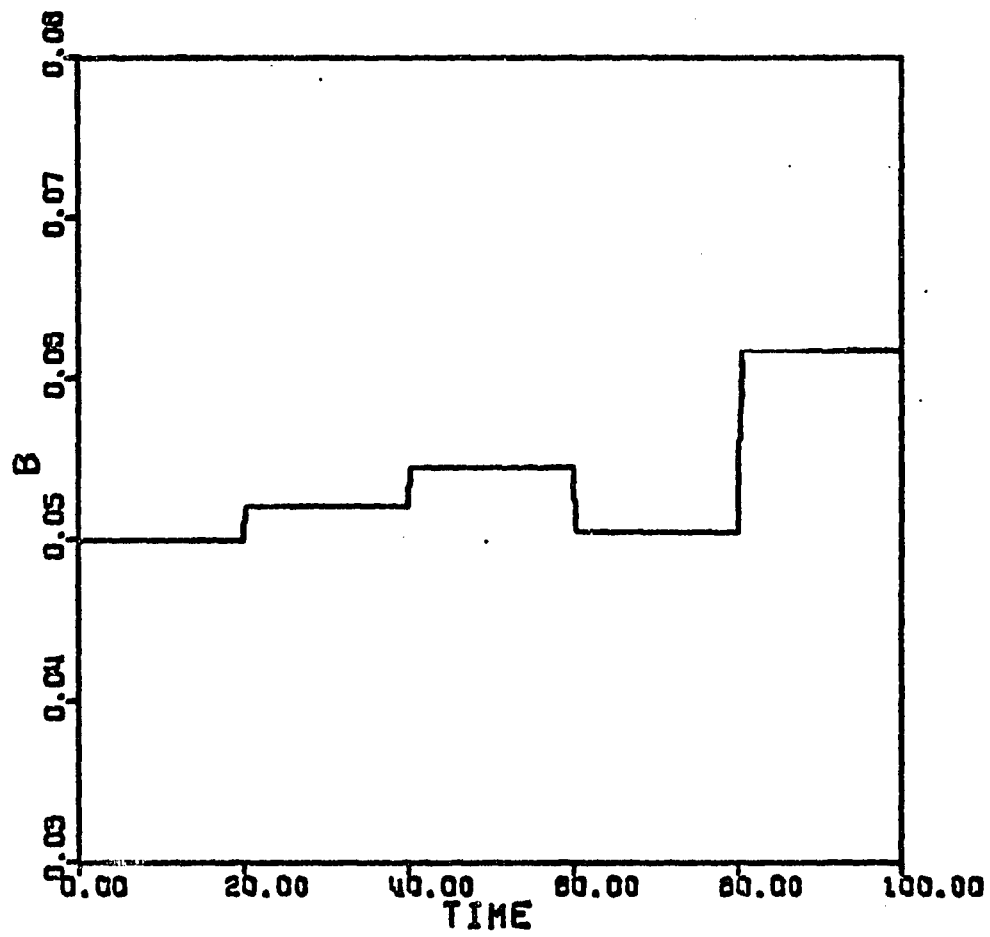


Figure 2.4b. Variation in Model Parameter B.

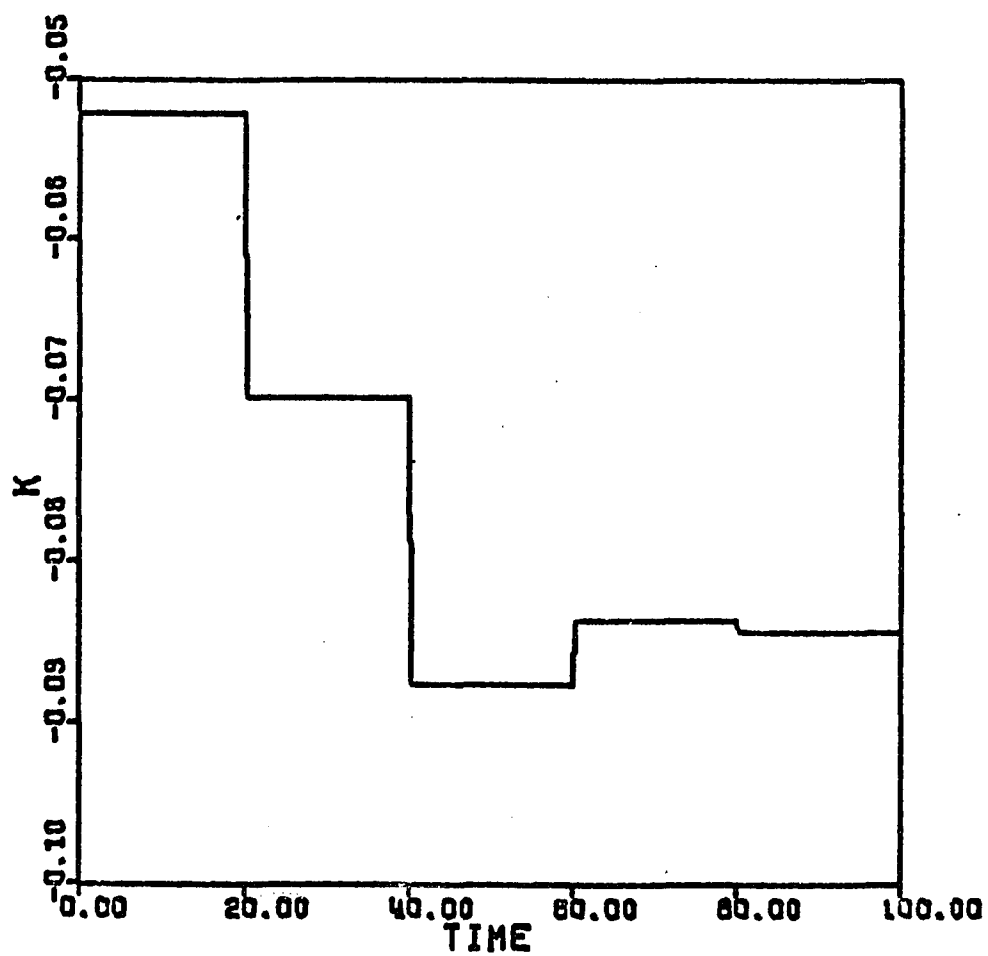


Figure 2.4c. Variation in Model Parameter K .

where T is the sampling time. The next step is to divide numerator and denominator by an appropriate factor so that the sum of the coefficients of $P(z)$ equals one. The controller transfer function, $D^*(z)$ is then found from the relationship

$$D^*(z) = \frac{Q^*(z)}{1-P^*(z)} \quad (2.25)$$

The main drawback of using the time-optimal approach is that essentially two controllers must be utilized. During the transition from one state to another, the output is held at one of the extremes as determined from either the switching times or alternatively from the switching curve. After the third switch is made a feedback control law is then implemented. Kalman's algorithm could be used at all times.

The only problem, however, is that Kalman's derivation assumed no limits on the control or manipulated variable. In the reactor problem as in most process applications some physical limitations will be placed on the system. Thus, when the sampling times are long, the range for the control variable will be small and Kalman's algorithm can be used directly. The steady-state response will be reached in two sampling instants but its cost will be somewhat larger than the optimal bang-bang case. If the sampling time is decreased, Kalman's algorithm computes values of the manipulated variable that exceed the limits. Simply replacing the computed value of the manipulated variable by the limiting values leads to an unstable algorithm. Some improvement

is obtained if the following procedure is used:

1. Calculate the control error as the difference between the set point and the controlled variable.
2. Calculate the manipulated variable as usual.
3. If the calculated manipulated variable exceeds the limit, replace its value by the limit.
4. Back-calculate from the algorithm equation the value of the control error that would have given the limiting value of the manipulated variable. Store this in place of the value calculated in Step 1. While this works at least in some cases, it can potentially produce a limit cycle in the output.

This technique can be demonstrated by controlling the backmix reactor using Kalman's algorithm. First, the values in the algorithm have to be determined from the plant model. For the second order system

$$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2.26)$$

the values for the gain and two time constants are already known from the sensitivity analysis or any other available means used to determine those parameter values. From equation 2.24

$$HG^*(z) = \left[\frac{z - 1}{z} \quad \frac{H(s)}{s} \right] * \quad (2.27)$$

or

$$HG^*(z) = \left[\frac{z-1}{z} \frac{K/\tau_1\tau_2}{s(s + 1/\tau_1)(s + 1/\tau_2)} \right]^* \quad (2.28)$$

From a table of z-transforms, this expression can be put entirely in the z-domain.

$$HG^*(z) = K(z-1) \left[\frac{1}{z-1} + \frac{\tau_1}{(\tau_2 - \tau_1)(z - e^{-T/\tau_1})} - \frac{\tau_2}{(\tau_2 - \tau_1)(z - e^{-T/\tau_2})} \right] \quad (2.29)$$

By multiplying the terms together and combining the coefficients of the different powers of z, the above equation can be put into the form:

$$G^*(z) = \frac{A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}} = \frac{P^*(z)}{Q^*(z)} \quad (2.30)$$

From the development of this algorithm all the coefficients must be normalized and this will force $\sum p_i = 1$. This normalization will produce new coefficients for the z terms. Next the controller algorithm is found from

$$D^*(z) = \frac{Q^*(z)}{1 + P^*(z)} = \frac{M(z)}{E(z)} \quad (2.31)$$

The final control equation can be written as

$$M_N = \beta_0 E_N + \beta_1 E_{N-1} + \beta_2 E_{N-2} + \alpha_1 M_{N-1} + \alpha_2 M_{N-2} \quad (2.32)$$

A more detailed development for these coefficients can be found in the appendix. As can be seen, the manipulated variable is calculated from the present and last two error signals along with the last two values of the manipulated variable. However, as noticed in the step by step procedure, the calculated value might be outside the allowed limits for the manipulated variable. In this case, the value of the manipulated variable is replaced by that limit and equation 2.32 is rearranged and solved for a new error signal. This pseudo error signal will be stored and used in the control algorithm for the next calculation. Also, the limit of the manipulated variable will be stored and not the calculated value.

Figure 2.6 compares the performance of Kalman's algorithm executed in this manner to the minimum time control law. Note that Kalman's algorithm overshoots somewhat, and does not perform as well.

Figure 2.6 also compares the performance of the two controllers described above to that of a conventional PI controller. Since the manipulated variable is constrained, reset windup is a potential problem, which can be largely circumvented by using the velocity form of the algorithm. The tuning parameters (proportional gain and reset time) in the PI controller used for Figure 2.6 were determined using Pattern Search to minimize the performance function ITAE (integral of time and absolute error) directly on the

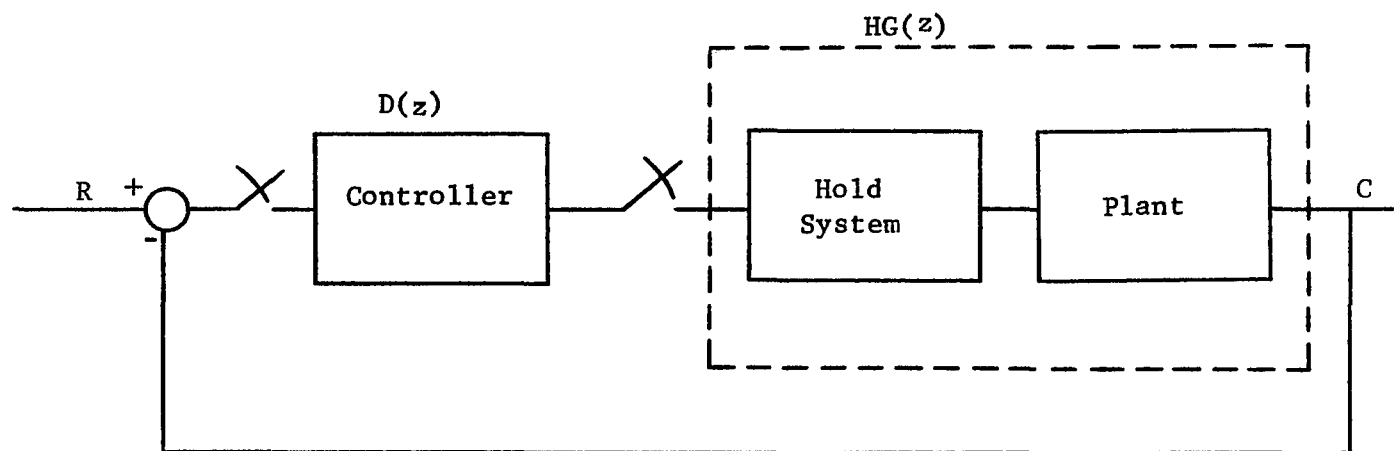


Figure 2.5. Typical sampled-data control system.

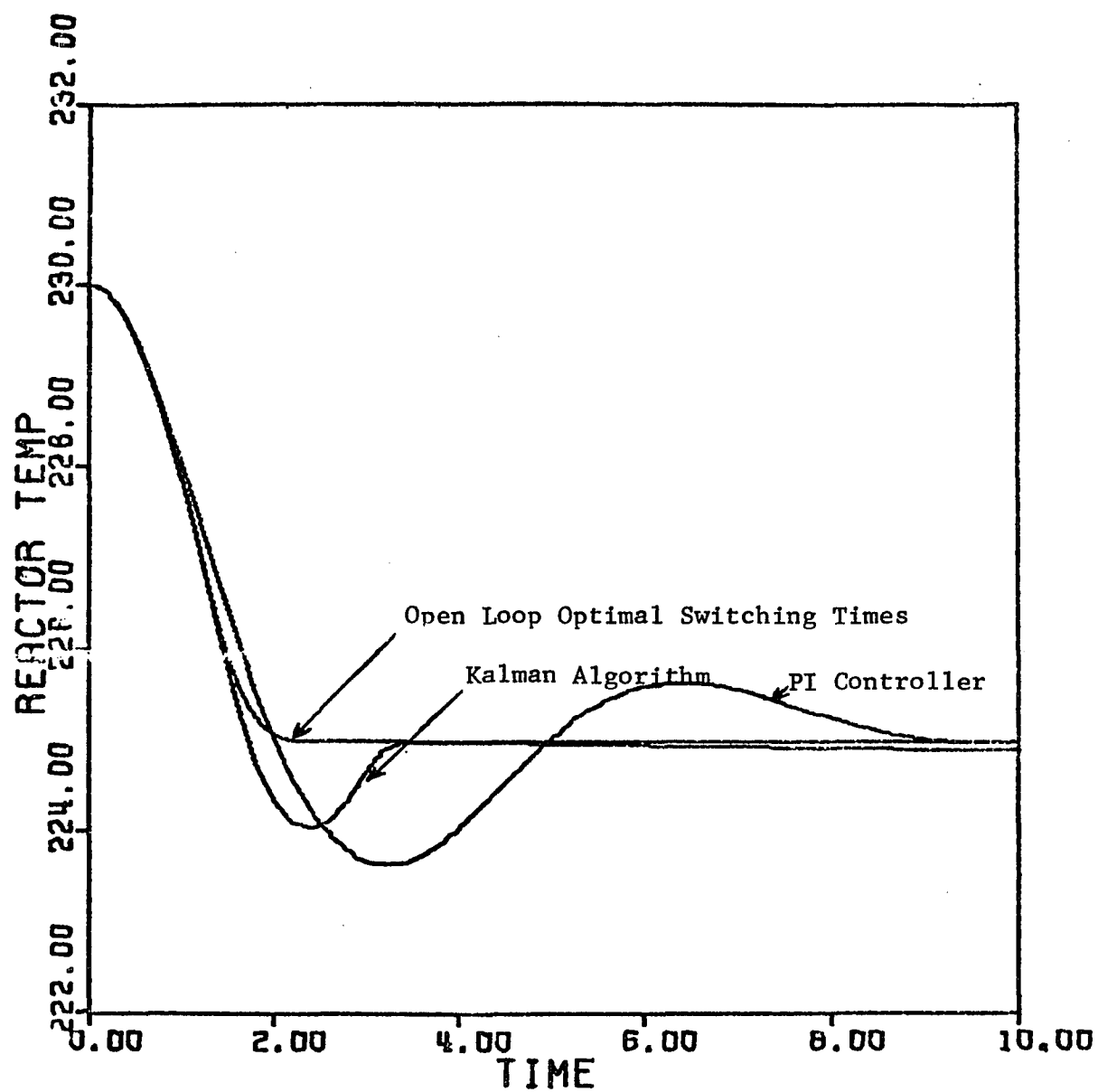


Figure 2.6. Temperature Response for Set Point Change Using Parameters from Sensitivity Analysis and Three Different Control Strategies

reactor itself. Thus, this illustrates the ultimate performance attainable by a PI controller, which is substantially poorer than both of the other two methods.

Summary

The use of sensitivity coefficients for process control applications offers several important advantages. They are well-suited for processes that have constantly changing parameters because the sensitivity coefficients will update the plant model resulting in an adaptive configuration. As long as the changes occur at a rate slower than the process time constant, the control system will be able to adjust itself in order to produce the optimum responses. Also, there would be no danger of the system changing in such a way that the controller would not be able to handle the situation and an unstable response would result. A non-computational advantage of the sensitivity method may result in the investigator getting more of a feel for the importance of certain parameters and the lack of importance of others by observing the values of the coefficients during the process response. However, the sensitivity coefficients do have several limitations. First, the starting points or initial guesses have to be relatively close to the true values or else instability might result. Also, because the solution in some regions becomes almost a gradient technique, the solution oscillates between parameter values. But both of these disadvantages may be circumvented by special techniques and the method can be applied successfully in almost all cases.

Nomenclature

A, B, K	Model parameters
c	System output
\hat{c}	Model output
Δc	Difference between system and model outputs
$D^*(z)$	Controller transfer function for sampled-data system
$F(s), f(t)$	System forcing function
$G(s)$	Model or plant transfer function
$HG(z)$	Plant and hold transfer function
$P(z)$	Numerator terms for plant and hold transfer function
$Q(z)$	Denominator terms for plant and hold transfer function
q_i	System parameters
Δq_i	Changes in system parameters
r	System input
s	Differential operator
T	Sampling time
t_2, t_4	Switching times for bang-bang controller
z	z -transform operator
α, β	Controller coefficients for Kalman algorithm
δ	Damping ratio
λ	Conversion factor for Marquardt's method

μ_i	Sensitivity coefficient
τ_1, τ_2	System time constants
ω_N	Natural frequency of system

Literature Cited

1. Kalman, R. E., from discussion in "Sampled-Data Processing Techniques for Feedback Control Systems" by Bergen and Ragazzini, AIEE Transactions, November 1954, pp. 245-246.
2. Latour, P. R., L. B. Koppel and D. R. Coughanowr, "Time-Optimum Control of Chemical Processes for Set-Point Changes", I&EC Process Design and Development, Vol. 6, No. 4, Oct. 1967, pp. 452-459.
3. Marquardt, D. W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", J.S.I.A.M., Vol. II, June 1963, pp. 431-441.
4. Meissinger, H. F., "The Use of Parameter Influence Coefficients in the Computer Analysis of Dynamic Systems", Proceedings of the Western Joint Computer Conference, May 1960.
5. Rose, R. E., Ph.D. Dissertation, University of Iowa, 1967.
6. Tomovic, R., Sensitivity Analysis of Dynamic Systems, McGraw-Hill Book Company, Inc., New York, 1963
7. Watson, J. W., "Sensitivity Functions in an Adaptive Control System and Their Application to Advanced Ballistic Reentry Systems," Ph.D. Dissertation, University of California, 1966.
8. Wilde, D. J., and C. S. Beightor, Foundations of Optimization, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968.

Chapter III

TUNING CONTROLLERS USING OPTIMAL CONTROL THEORY

One disadvantage of the tuning techniques currently available is that the performance objective (quarter decay ratio, minimum integral of absolute error, or others) is specified in the development of the technique and cannot be varied at the discretion of the user. In this chapter, optimal control theory is used to develop a tuning technique based on a performance specification that contains a parameter that the user can vary to "tighten-up" or "loosen" the setting produced by the technique. Starting with a first-order lag plus dead time process model, the first-order Pade approximation is substituted for the dead time, producing a result that can be expressed as a linear state equation. Applying state regulator theory with a quadratic performance index produces the settings for the conventional proportional-integral-derivative (PID) control law. This chapter considers both the set point (servo) and load (regulator) control problems, and extends the results to discrete systems.

General Process Control Problem

In order to simplify the calculations, tuning methods are generally based on relatively simple process models. For many processes a suitable model is the first-order lag plus dead

time, whose transfer function is:

$$G_p(s) = \frac{c(s)}{u(s)} = \frac{K e^{-\theta s}}{\tau s + 1} \quad (3.1)$$

where

$c(s)$ = system output

$u(s)$ = system input

K = system gain

τ = system time constant

θ = system dead time (transportation lag or time delay)

s = Laplace transform variable

Although the accuracy of this model is not especially good in many cases, considerable extra effort is required to develop higher order models. In the typical feedback control scheme, the output c is controlled by comparing it with some setpoint r and then manipulating the input u . The physical configuration for this system is illustrated in Figure 3.1. A common and easily implemented controller is the PID controller whose transfer function is:

$$G_c(s) = K_c \left(1 + \frac{1}{T_i s} + T_d s \right) = \frac{u(s)}{c(s)} \quad (3.2)$$

where

K_c = controller gain

T_i = reset time or integral time

T_d = rate time or derivative time

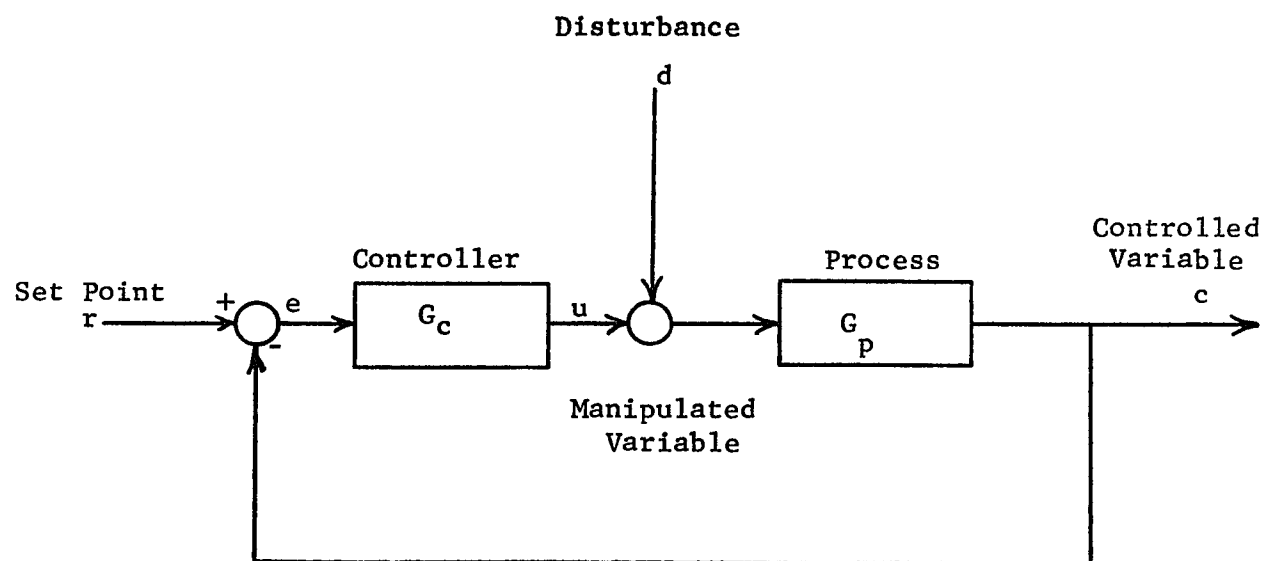


Figure 3.1.
General Process Control Problem

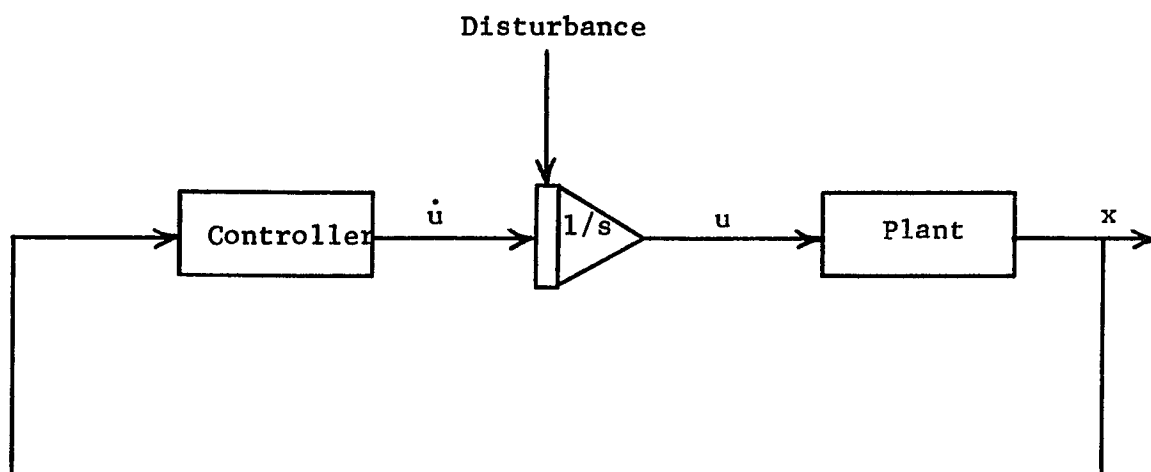


Figure 3.2.
Alternate Representation of the Disturbance

An alternate approach consists of removing the dead time from the closed loop by means of a compensator such as the Smith predictor (11). The major disadvantage is that the result is not a simple feedback control law and cannot be implemented with conventional analog hardware.

The Optimal Control Formulation

In order to obtain controller parameters with optimal control theory, the process equations must first be put into state variable form:

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \quad (3.3)$$

where

\underline{x} is an n-dimensional state vector

\underline{u} is an r-dimensional control vector

\underline{A} is an n x n system matrix

\underline{B} is an n x r control matrix

The problem of interest is the classical state regulator problem, which uses the quadratic performance index

$$J = \int_{t_0}^{t_f} (\underline{x}^T \underline{Q} \underline{x} + \underline{u}^T \underline{P} \underline{u}) dt \quad (3.4)$$

The \underline{Q} matrix is symmetric and positive semidefinite while the \underline{P} matrix is symmetric and positive definite. The solution to this problem can be found by solving the matrix Riccati

equation

$$\dot{\underline{R}} = -\underline{R} \underline{A} - \underline{A}^T \underline{R} + \underline{R} \underline{B} \underline{P}^{-1} \underline{B}^T \underline{R} - \underline{Q} \quad (3.5)$$

As t_f approaches infinity, $R(t)$ becomes constant. Therefore, $\dot{\underline{R}}$ becomes zero and the reduced or degenerate Riccati equation results:

$$\underline{A}^T \underline{R} + \underline{R} \underline{A} - \underline{R} \underline{B} \underline{P}^{-1} \underline{B}^T \underline{R} + \underline{Q} = 0 \quad (3.6)$$

The above equation is a set of nonlinear algebraic equations which can be solved for the elements of \underline{R} . Since \underline{R} is required to be positive definite, this leads to the optimal control law

$$\underline{u} = \underline{K} \underline{x} \quad (3.7)$$

where \underline{K} is a constant matrix defined by

$$\underline{K} = -\underline{R}^{-1} \underline{B}^T \underline{P} \quad (3.8)$$

Approximating the Dead Time

If the time delay in the first order plus dead time system of equation (3.1) is replaced by a first order Padé approximation, the state equation 3.3 can represent the resulting approximation. The Padé approximation substitutes $(1-\theta s/2)/(1+\theta s/2)$ for the $e^{-\theta s}$ term, yielding the new transfer function:

$$G_p(s) = \frac{K}{\tau s + 1} \frac{1 - \theta s/2}{1 + \theta s/2} \quad (3.9)$$

The differential equation counterpart to this transfer function is:

$$\frac{\theta\tau}{2} \ddot{c} + \left(\tau + \frac{\theta}{2}\right) \dot{c} + c = K u - \frac{K\theta}{2} \dot{u} \quad (3.10)$$

Equation 3.10 can be represented by state variables as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-2}{\theta\tau} & \frac{-\theta+2\tau}{\theta\tau} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{\tau} \\ \frac{\theta+4\tau}{\theta\tau^2} \end{bmatrix} \begin{bmatrix} K u \end{bmatrix} \quad (3.11)$$

where

$x_1 = c$, the system output

$x_2 = \dot{c} + u/\tau$

Notice that the control in the above equations is really equal to the system input multiplied by the process gain, K . The optimal control for the above system and the quadratic cost index $J = \int (\underline{x}^T Q \underline{x} + p u^2) dt$ will be determined with the solution of the Riccati equation. This control law is of the form $u = \underline{K} \underline{x}$, and as can be seen from the definitions of x_1 and x_2 , will become a proportional plus derivative controller.

This formulation has not considered the unmeasured disturbance in the block diagram of Figure 3.1. For step changes in the disturbance, redefining the physical system as illustrated in Figure 3.2 permits state variable notation to be used for this term also (Ref. 3).

The state equation for this formulation becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1/\tau \\ -2/\theta\tau & -\frac{\theta+2\tau}{\theta\tau} & \frac{\theta+4\tau}{\theta\tau^2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \quad (3.12)$$

where

$$x_3 = u + d$$

$$v = \dot{x}_3 = \dot{u}$$

$$d = \text{system disturbance}$$

$$x_1 = c, \text{ system output}$$

$$x_2 = \dot{c} + x_3/\tau$$

The initial conditions for system 3.12 are:

$$\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$

The disturbance d is considered to be constant and the Riccati equation is used to solve the classical state regulator problem yielding:

$$\dot{u} = K_1 x_1 + K_2 x_2 + K_3 x_3 \quad (3.13)$$

This equation can be integrated and the x_2 and x_3 terms eliminated by means of the process equations 3.12. From 3.12, the first equation gives

$$x_2 = \dot{x}_1 + x_3/\tau$$

and

$$(3.14)$$

$$\int x_2 = x_1 + \int (x_3/\tau) dt$$

Eliminating x_2 from the second process equation and solving for x_3

$$x_3 = \theta \tau \ddot{x}_1 + \theta \dot{u} + x_1 + (\theta + \tau) \dot{x}_1 \quad (3.15)$$

Equations 3.14 and 3.15 are substituted into 3.13 to obtain a control law in terms of x_1 , the system output:

$$u = \frac{1}{1 - (K_2/\tau + K_3)\theta} \left\{ \left[K_1 + K_2/\tau + K_3 \right] \int x_1 dt + \left[K_2 + (K_2/\tau + K_3)(\theta + \tau) \right] x_1 + \left[K_2/\tau + K_3 \right] \theta \tau \dot{x}_1 \right\} + K_0 \quad (3.16)$$

where K_0 is a constant of integration. In practice, K_0 is the value

of the controller output when the loop is placed on automatic.
 Since x_1 equals c ,

$$u = K_1' c + K_2' \int c dt + K_3' dc/dt + K_0 \quad (3.17)$$

The parameters K_1' , K_2' , and K_3' are readily related to the conventional tuning parameters K_c , T_i , and T_d . Nondimensionalizing the time by dividing by the time constant and removing the system gain from the process will cause the following equations to hold:

$$t' = t/\tau$$

$$u' = K u$$

$$\dot{u}' = K \tau \dot{u}$$

and the cost function becomes

$$\begin{aligned} J &= \int (x_1'^2 + p' \dot{u}'^2) dt' \\ &= \int (x_1'^2 + p K^2 \tau^2 \dot{u}^2) dt \end{aligned}$$

or

$$p' = p K^2 \tau^2 \quad (3.18)$$

Figure 3.3 gives the dimensionless responses to a step change in load for various values of p' . Note that small values of p' give tight control; large values of p' give loose control.

For various values of p' , Figures 3.4a-c relate the controller gain K_c , reset time T_i , and the derivative time T_d to the model parameters K , θ , and τ . To use these graphs, the practitioner follows this procedure:

1. Determine the model parameters by any technique he desires. The short cut method suggested by Miller (6) is recommended.
2. Use Figure 3.3 to obtain some estimate of the value of p' that is most appropriate to the application.
3. Determine the tuning parameters from the graphs in Figure 3.4.
4. Try these settings on the actual system. If the settings are too tight, increase p' and repeat from step 3. If the settings are too loose, decrease p' and repeat from step 3.

The main advantage of this approach over previous methods (2,5,12,14) is that the user has more control over the criterion function, whereas earlier methods were restricted to a specific criterion function such as the one-quarter decay ratio or integral of the absolute error.

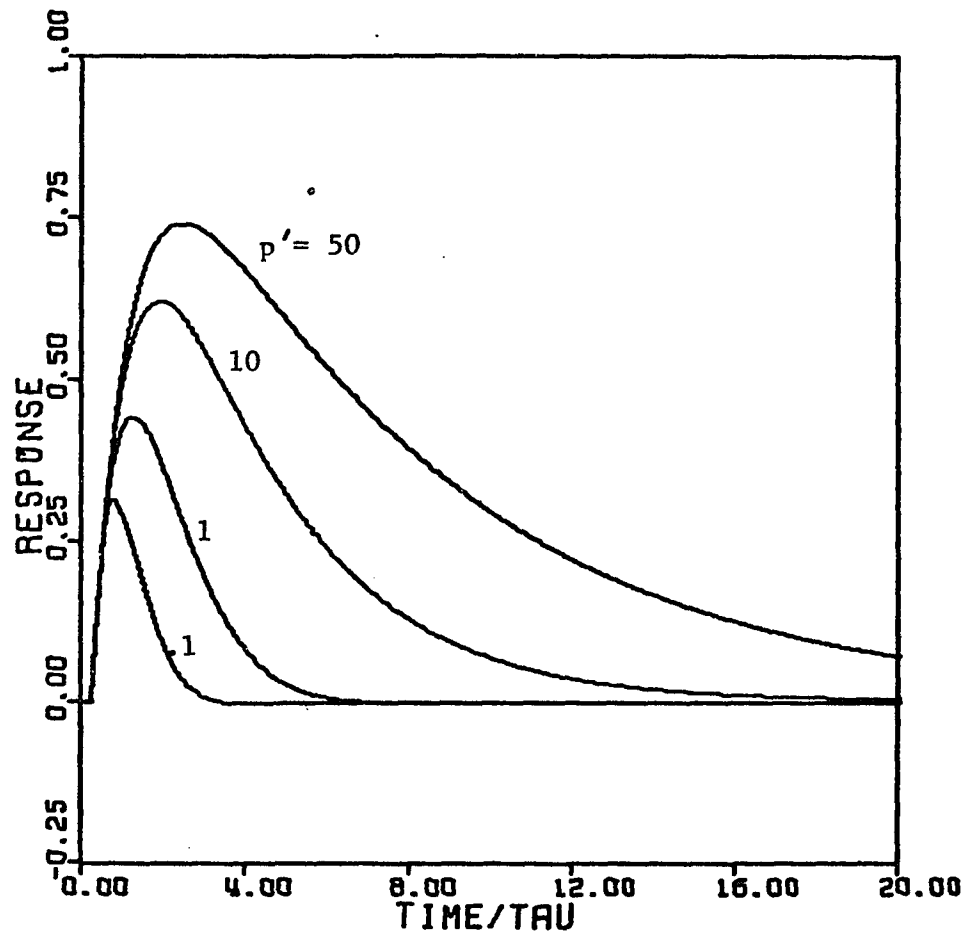


Figure 3.3.

Dimensionless Responses to a Step
Change in Load for Various Values of p'

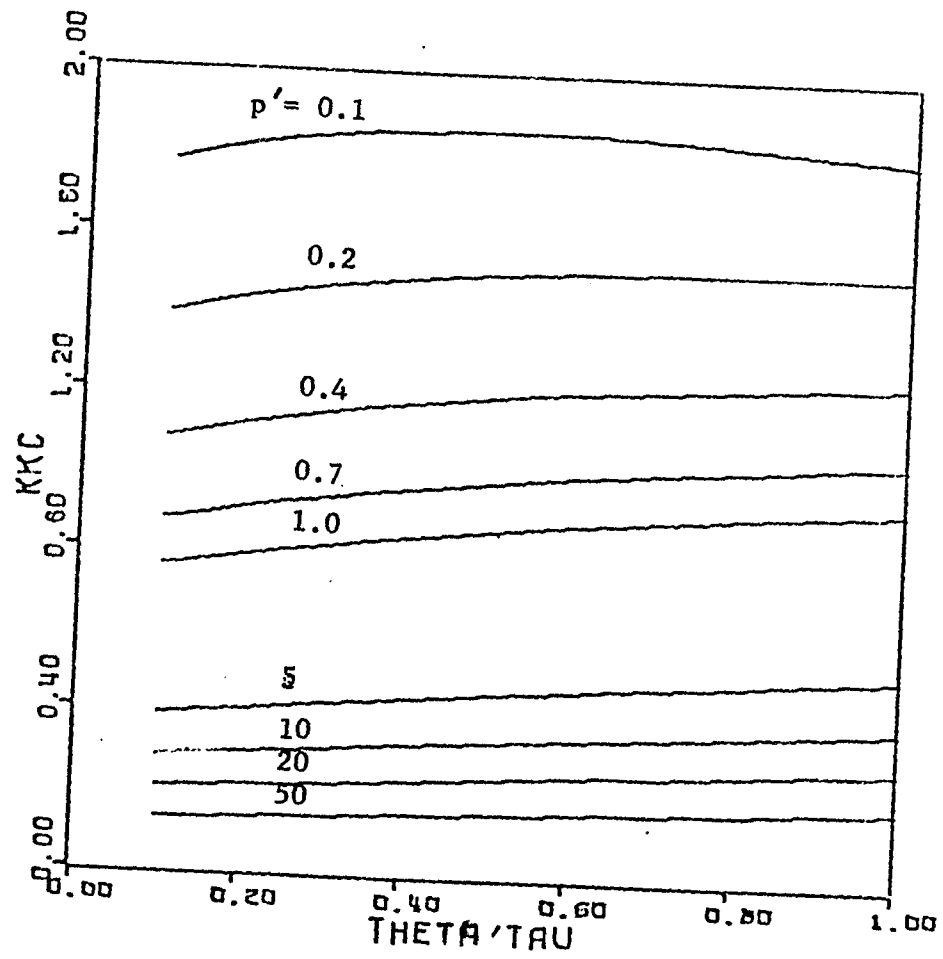


Figure 3.4a. Optimum Control Parameters For Various Values of p'

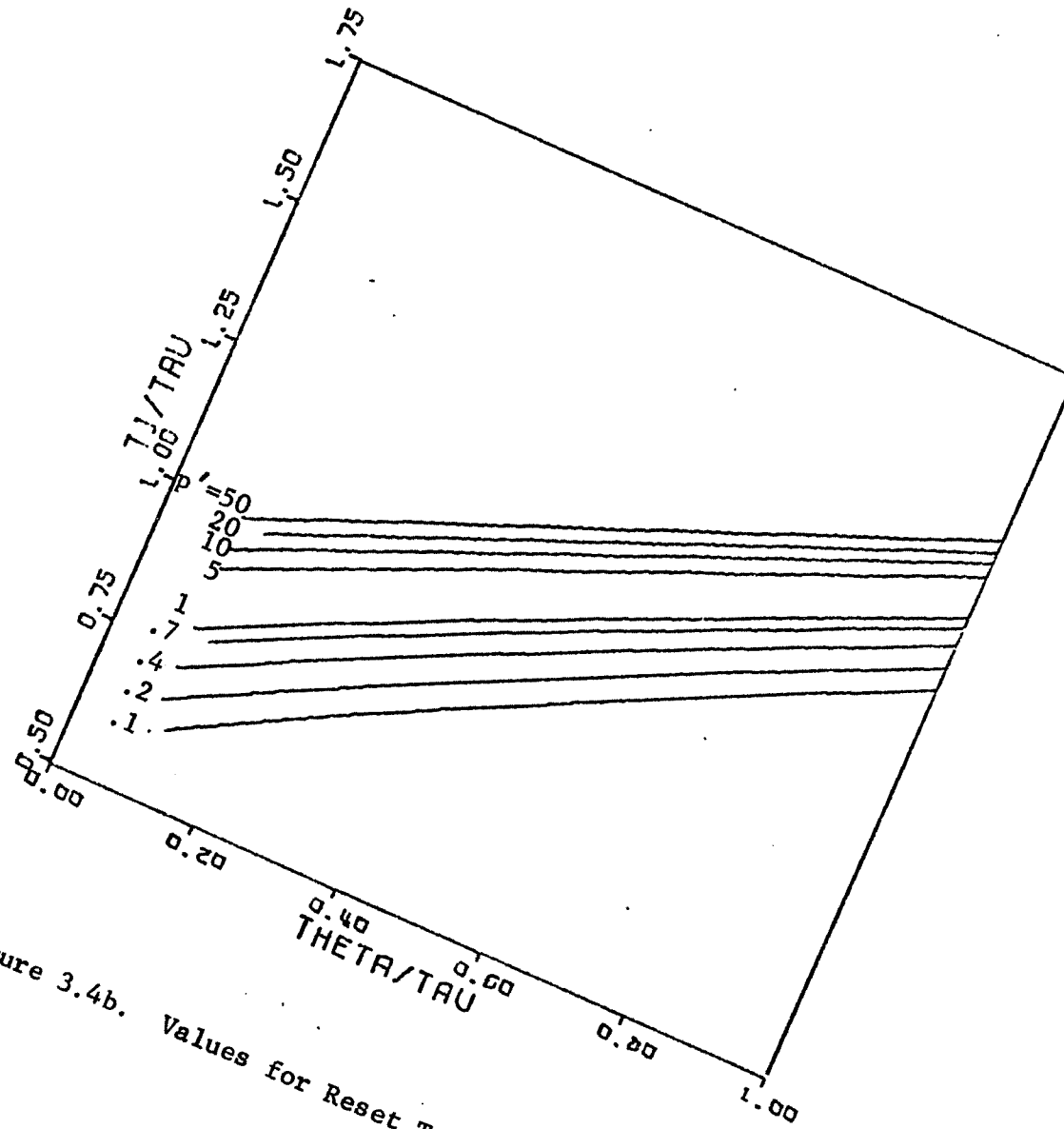


Figure 3.4b. Values for Reset Time.

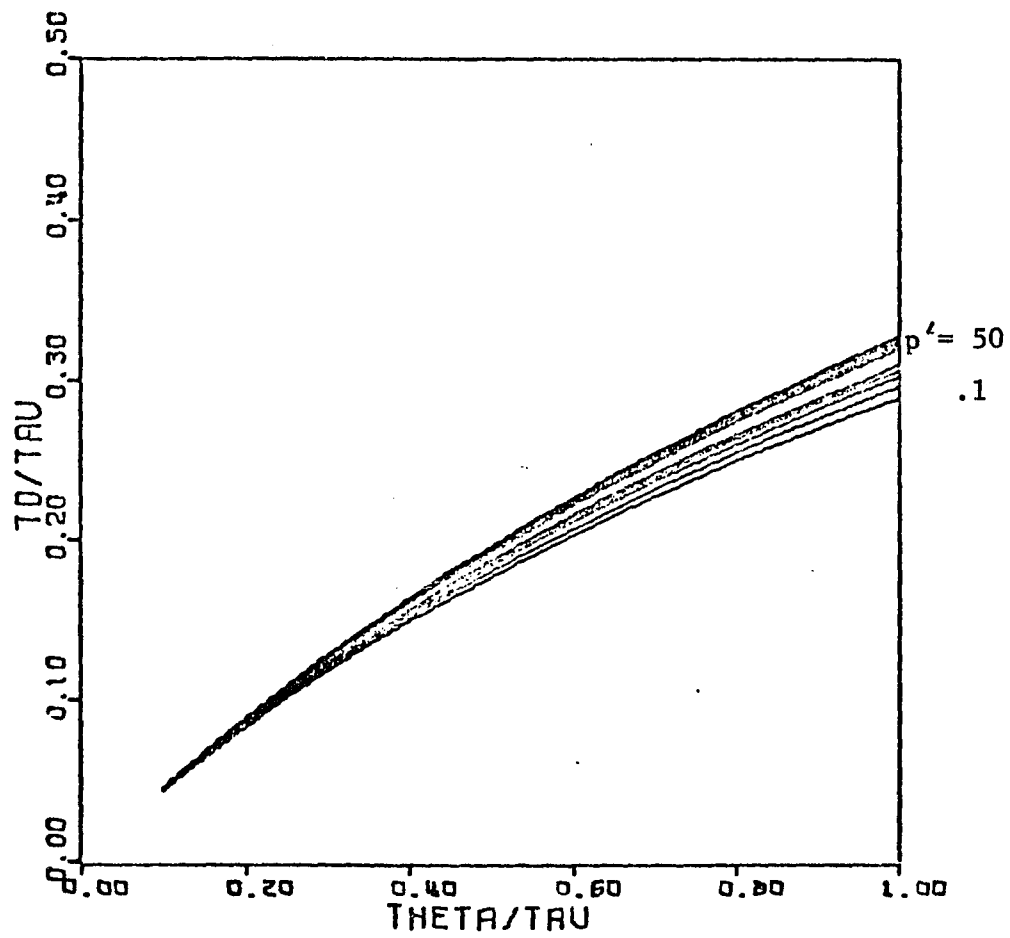


Figure 3.4c. Values for Rate Time.

Verification of the Method

To test the results on a realistic system, the stirred chemical reactor in Figure 3.5 was considered. The reactor temperature T_p is controlled using the jacket water flow rate m as the manipulated variable. In modeling the open-loop response of the system to a step change in m (see Figure 3.6a), the three parameters in a first order lag plus dead time model were determined by means of least squares fit to be:

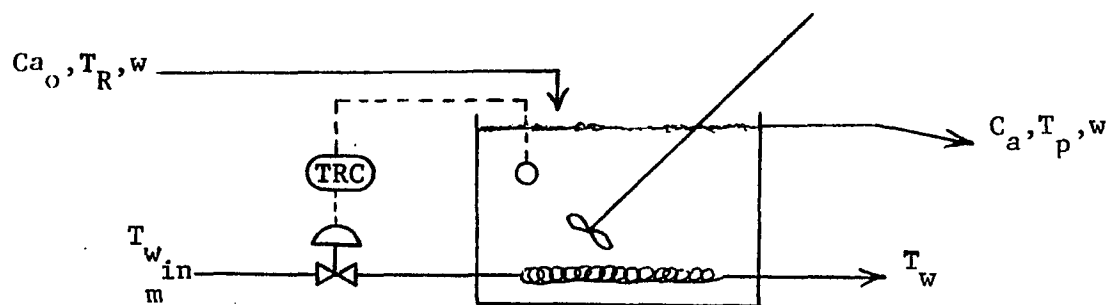
$$K = \text{system gain} = -0.01^\circ\text{F/lb}$$

$$\tau = \text{time constant} = 12.6 \text{ min}$$

$$\theta = \text{dead time} = 2.4 \text{ min}$$

Essentially the same results are obtained using the graphical approach commonly used in other controller tuning techniques. For a selected value of p' , the tuning parameters can be determined from Figure 3.4.

Both load and set point changes were investigated. For load changes the flow rate of reactants to the reactor was increased from 1200 lb/min for 50 minutes and then returned to the steady state value of 1000 lb/min. Figure 3.6b shows the open loop responses to these changes. Note the initial "dip" in the temperature. For set point changes the set point was reduced to 185°F and then increased back to 190°F after 50 minutes.



Reaction Kinetics: $2A \rightarrow B$ $r_a = -kC_a^2$ $k = k_o \exp(-a/\tau)$

System eqns:

Unsteady state enthalpy balance on kettle:

$$w C_p (T_R - T_p) + (-\Delta H) k C_a^2 V_t - UA (T_p - T_w) = V_t \rho_t C_p (dT_p/dt)$$

Unsteady state enthalpy balance on jacket:

$$m(T_{win} - T_w) - UA(T_w - T_p) = M_c (dT_w/dt)$$

Unsteady state balance on component A in the kettle:

$$w(Ca_o - C_a)/\rho_t - V_t k C_a^2 = V_t (dC_a/dt)$$

Steady State Conditions

$$\begin{aligned} T &= 190^\circ\text{F} \\ k &= .0278 \text{ f}^3/\text{lb} \cdot \text{min} \\ w &= 1000 \text{ lb/min} \\ T_R &= 150^\circ\text{F} \\ Ca_o &= 9 \text{ lb/f}^3 \\ C_a &= 3.6 \text{ lb/f}^3 \\ T_{win} &= 80^\circ\text{F} \end{aligned}$$

System Parameters

$$\begin{aligned} C_p &= .9 \text{ BTU/lb} \cdot ^\circ\text{F} \\ V_t &= 250 \text{ f}^3 \\ \rho_t &= 60 \text{ lb/f}^3 \\ a &= 2560^\circ\text{R} \\ M_c &= 400 \text{ lb} \\ UA &= 600 \text{ BTU/min} \\ -\Delta H &= 867 \text{ BTU/lb} \end{aligned}$$

Figure 3.5. Chemical Reactor.

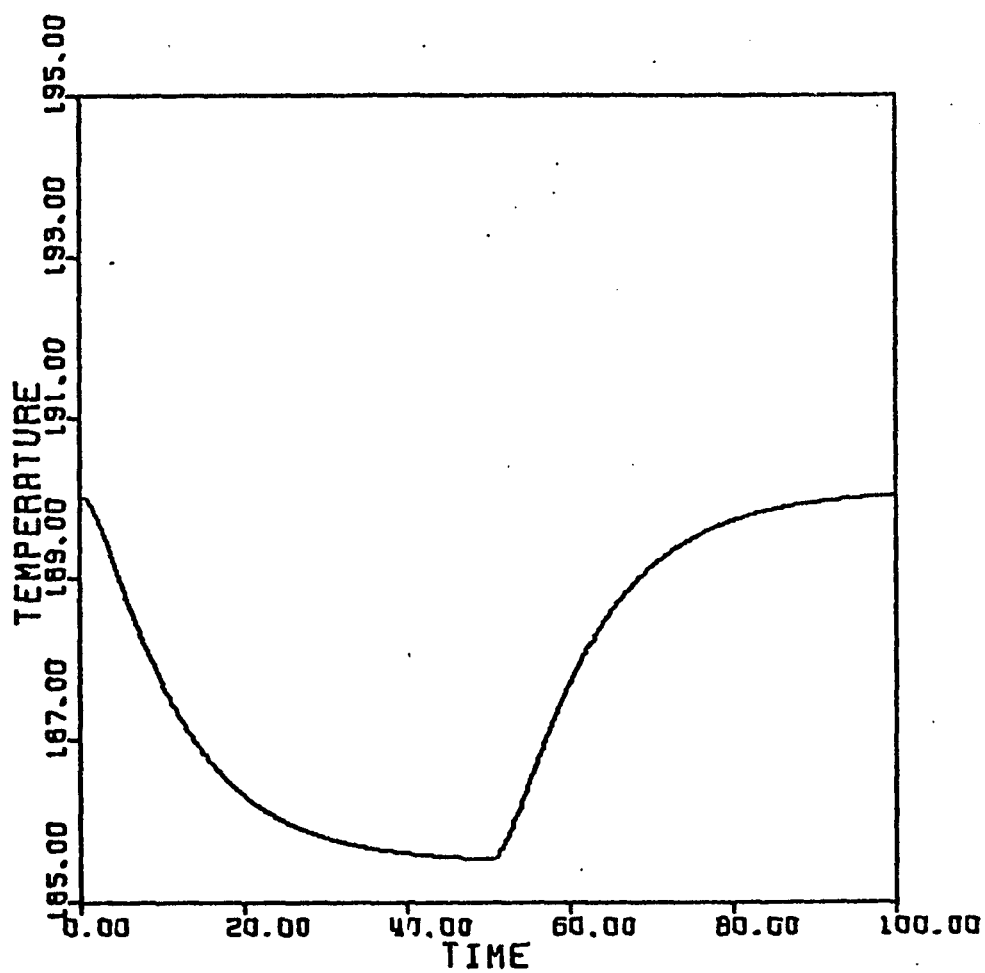


Figure 3.6a.

Open Loop Response to Step Change
in the Water Flow Rate

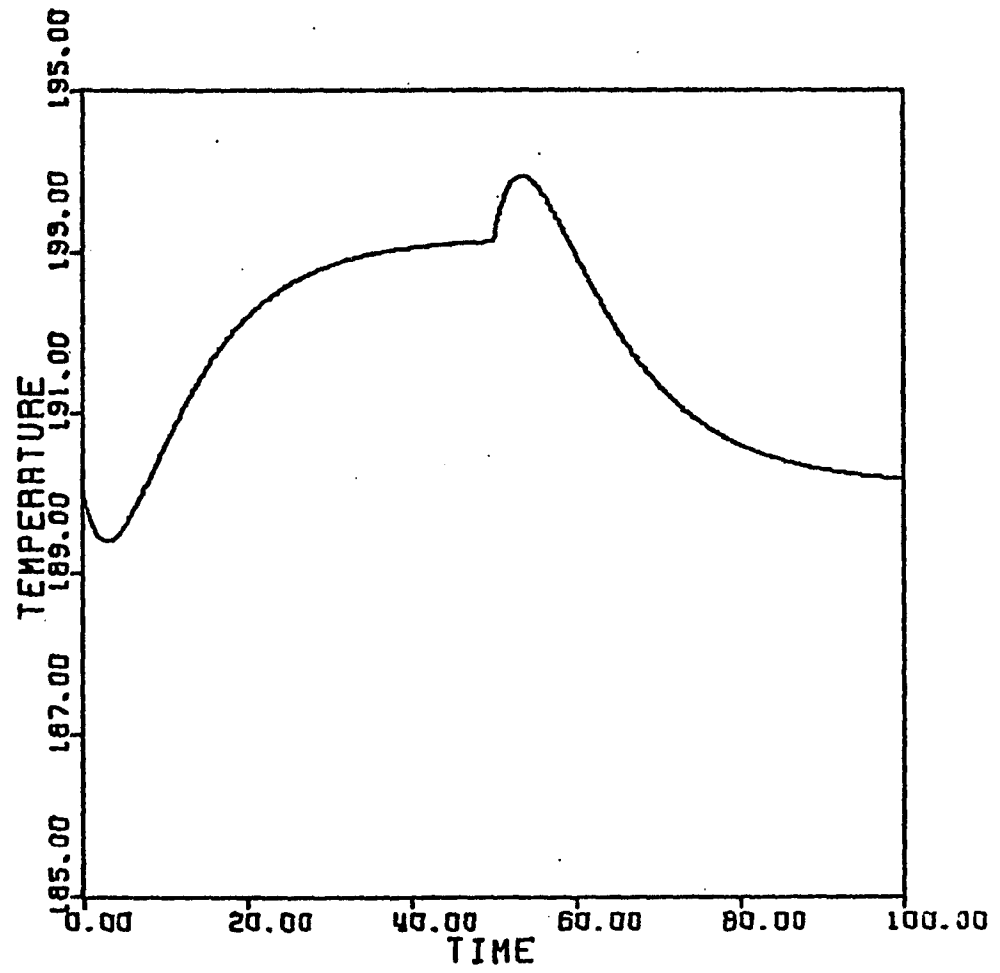


Figure 3.6b.

Open Loop Response to Change in Flow
Rate of Reactants

Figure 3.7 illustrates the performance of the controller tuned for various values of p' for both load changes and set point changes. The nonlinear nature of the system causes some differences between the responses to the first changes and the responses to the second changes. But for a given value of p' , note that the set point responses appear to be more oscillatory. For example, for p' equal to 10, the load responses are definitely overdamped, but the set point responses exhibit significant overshoot. Rovira (9) recognized the fact when he proposed different tuning relationships for set point responses than for load responses.

As developed, the tuning technique in this paper is for step changes in load. To apply these settings to set point responses, Rovira suggests that a modified form of the PID algorithm be used. The most frequently used form of the PID controller is given by the equation:

$$u(t) = K_c \left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (3.19)$$

In the modified algorithm, the feedback variable replaces the error signal in all modes except the integral mode. This algorithm, called the PCID is

$$u(t) = K_c \left[-c(t) + \frac{1}{T_i} \int e(t)dt - T_d \frac{dc(t)}{dt} \right] \quad (3.20)$$

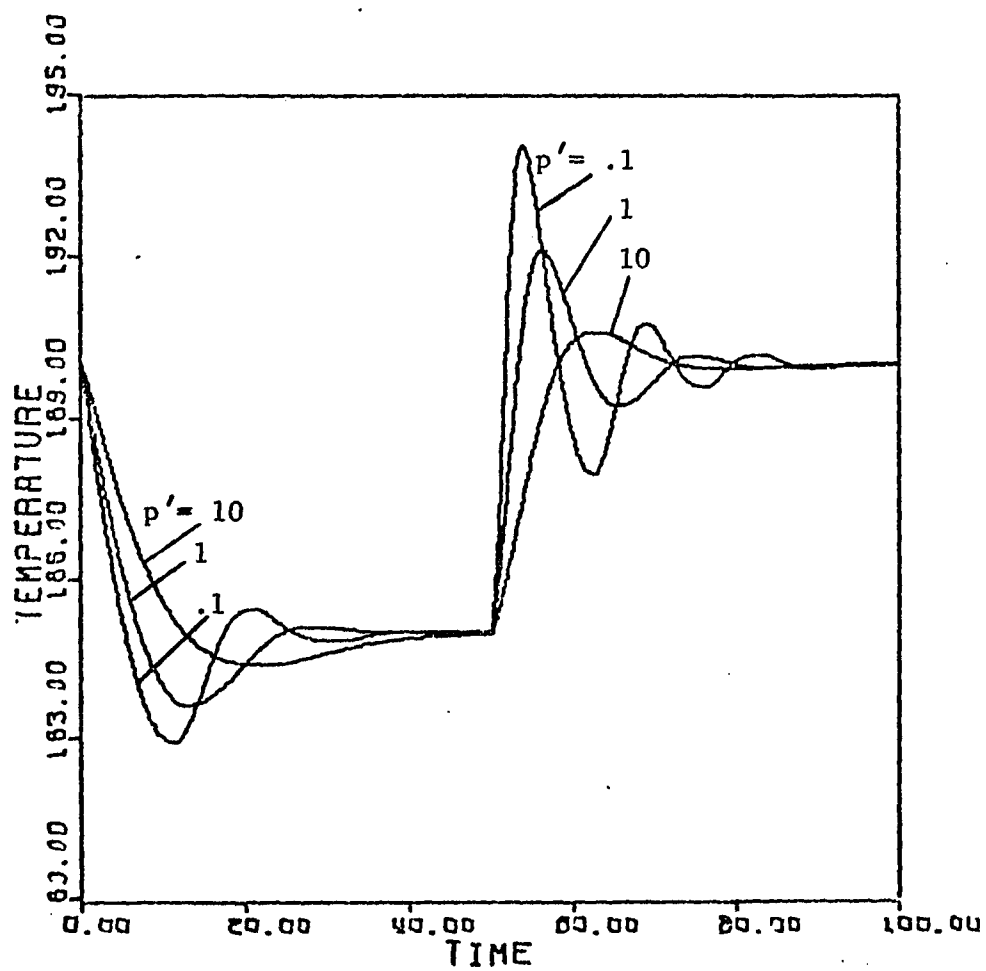


Figure 3.7a.

Responses for Setpoint Change .
With Continuous Feedback PID Controller

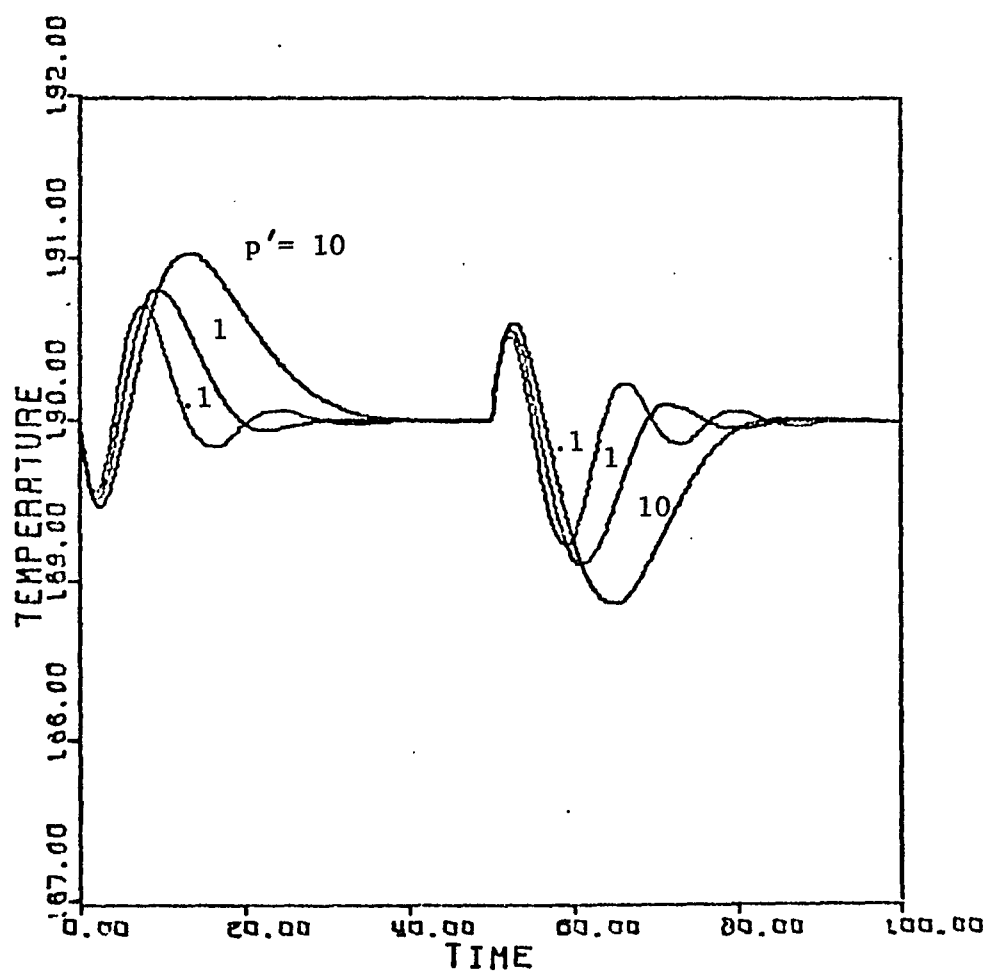


Figure 3.7b.

Responses for Load Change with
Both PID and PCID Continuous Controllers

Since r equals zero for load changes, $e(t)$ equals $-c(t)$, and equations 3.19 and 3.10 are identical. Consequently, the performance of these two algorithms is different only for set point changes.

For the PCID algorithm, the response to load changes is given by Figure 3.7b, since its performance for this case is identical to the standard PID algorithm. For set point changes, Figure 3.8 illustrates the performance of the PCID algorithm for various values of p' . Note that the qualitative shape of the response as the controlled variable approaches the set point is similar to that of the responses in Figure 3.7b. In other words, the PCID algorithm is more consistent than the PID algorithm.

However, note that the PID algorithm is capable of producing set point responses superior to that of the PCID (compare the response for $p' = 10$ in Figure 3.7a to any of those in Figure 3.8). The advantage of PCID is its consistency for both load and set point changes.

Sampled-data Systems

Although this method was presented for use with a continuous analog controller, it is easily adapted for use in sampled-data systems. A typical sampled-data control loop is shown in Figure 3.9, where the section containing the comparator and controller (dashed lines) would be a digital computer.

In this configuration the error signal e is available only at the sampling instants. The controller or digital computer

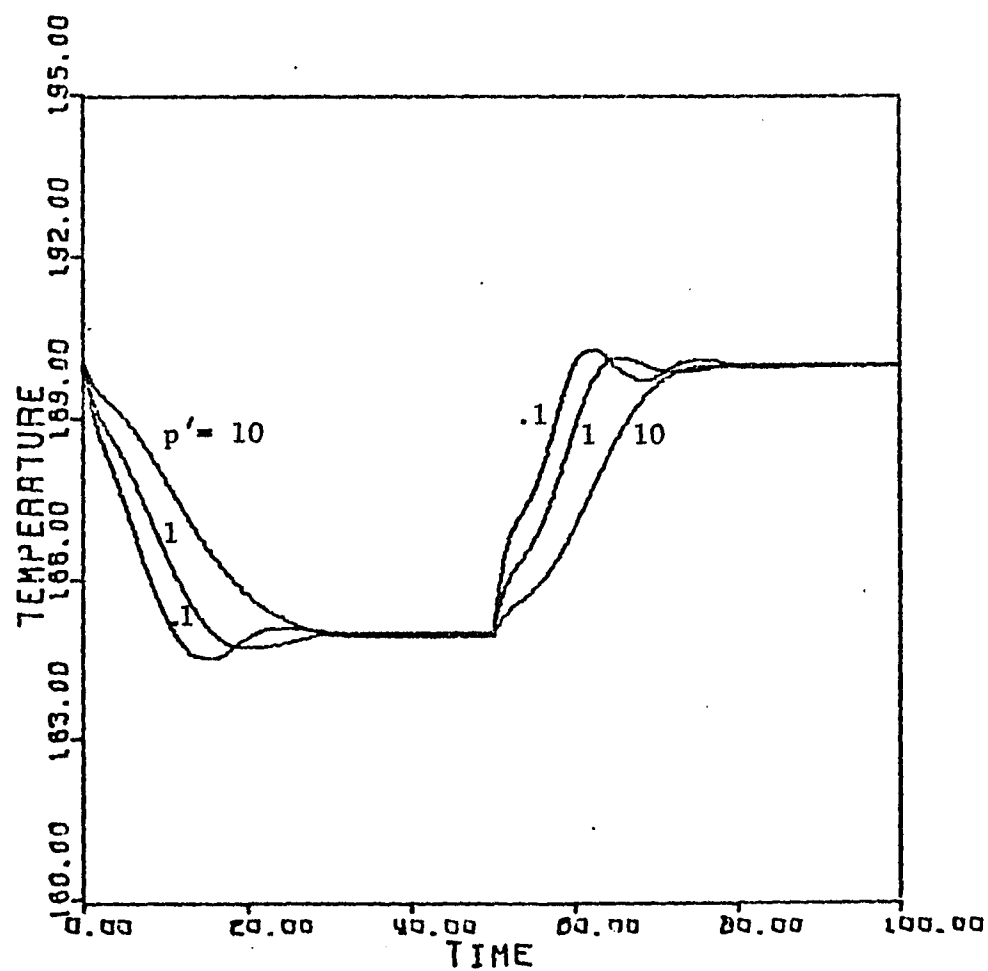


Figure 3.8.

Response for Setpoint Change
With Continuous PID Controller

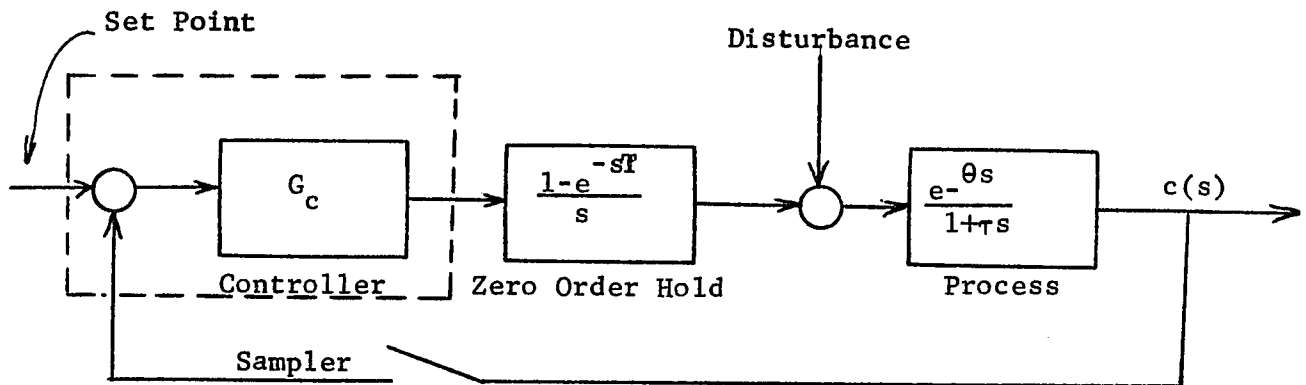


Figure 3.9a. Sampled-Data System

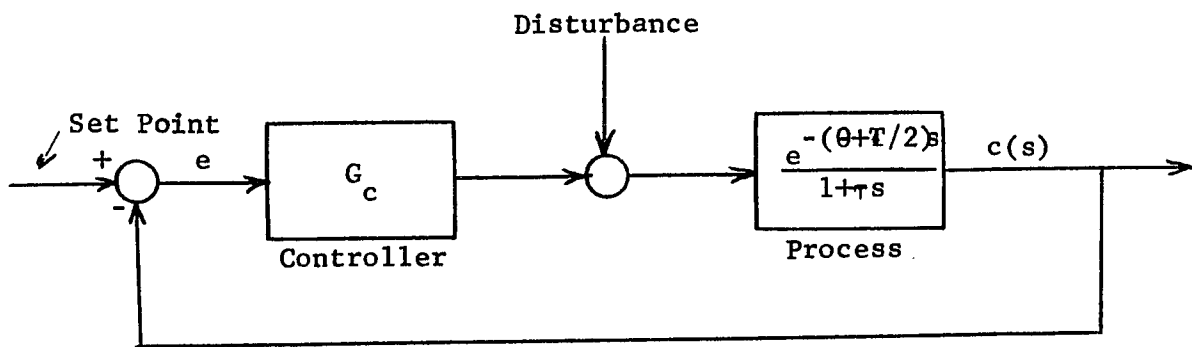


Figure 3.9b.

Continuous Approximation

Figure 3.9.

Dead Time Approximation to Sample and Hold

then calculates a new value for the manipulated variable which is also discrete and available only at sampling instants. Between sampling times a data hold is employed to keep the manipulated variable at the last value calculated. It can be shown that the discrete control system represented in Figure 3.9 can be successfully approximated by an equivalent continuous system, where a pure dead time of one-half the sampling time is added to the dead time of the process (8). Thus, the continuous system in Figure 3.9b is equivalent to the sampled-data system of Figure 3.9a. This approximation cannot be employed at very long sampling times approaching the system time constant or if the system is operating near its stability limits.

Using this method to approximate the discrete system with a continuous one, optimal control can then be applied to tune the PID or PCID controller. This was tested on the reactor system described earlier. The same model parameters were used to determine the tuning constants except that one-half of the sampling time was added to the system dead time. The system responses (Figure 3.10) for different values of the p' parameter were found to be comparable to those responses of the continuous system when the sampling rate was fast. However, as the sampling time increased, the response became more unstable and unacceptable. As would be expected, the dead time approximation degenerates as the sampling time approaches the system time constant, producing unsatisfactory control loop performance.

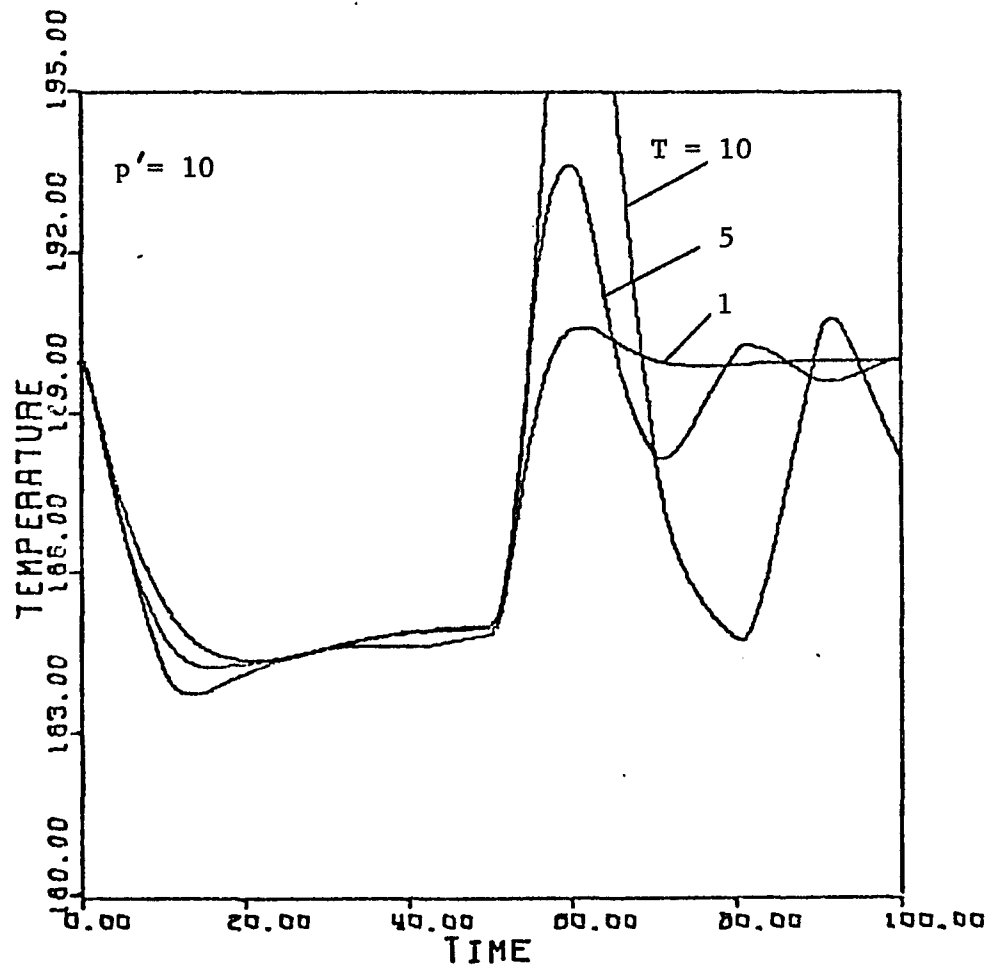


Figure 3.10a.

Responses for Setpoint Change with PID Controller
and Sampled-data System

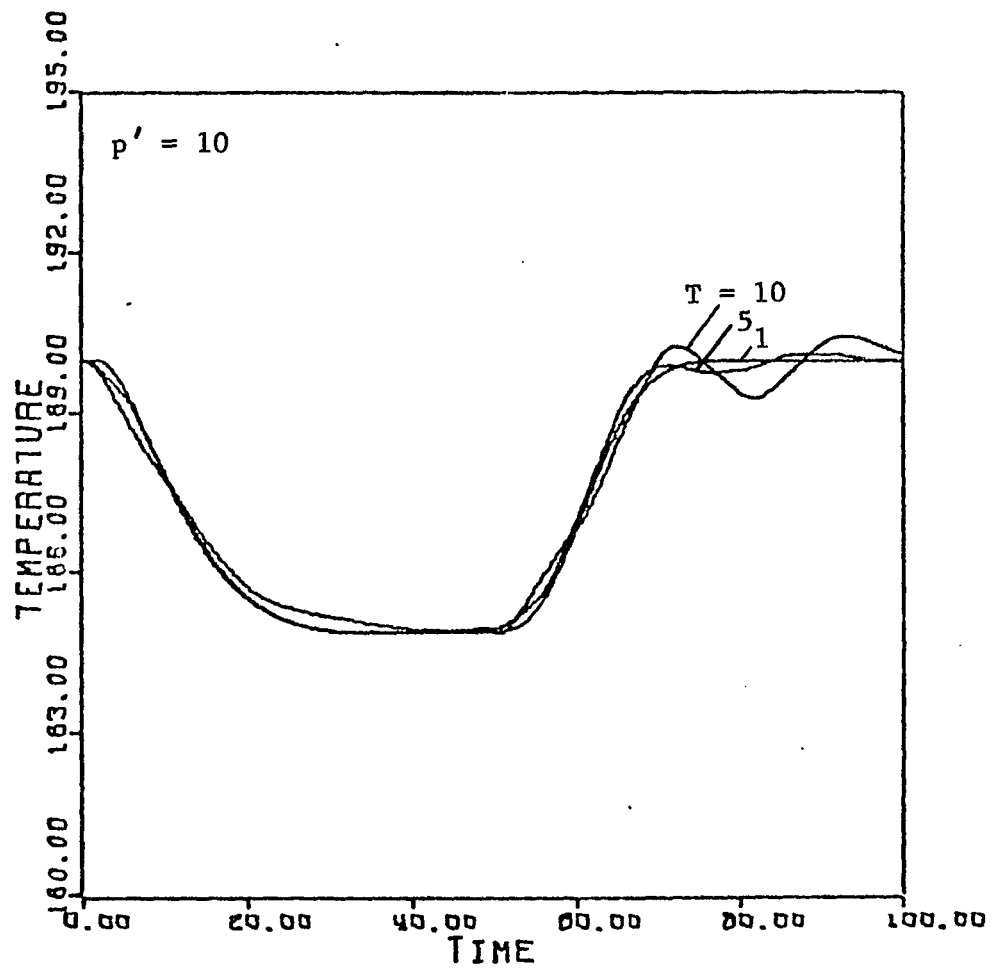


Figure 3.10b.

Response to Setpoint Change with PCID Controller
and Sampled-data System

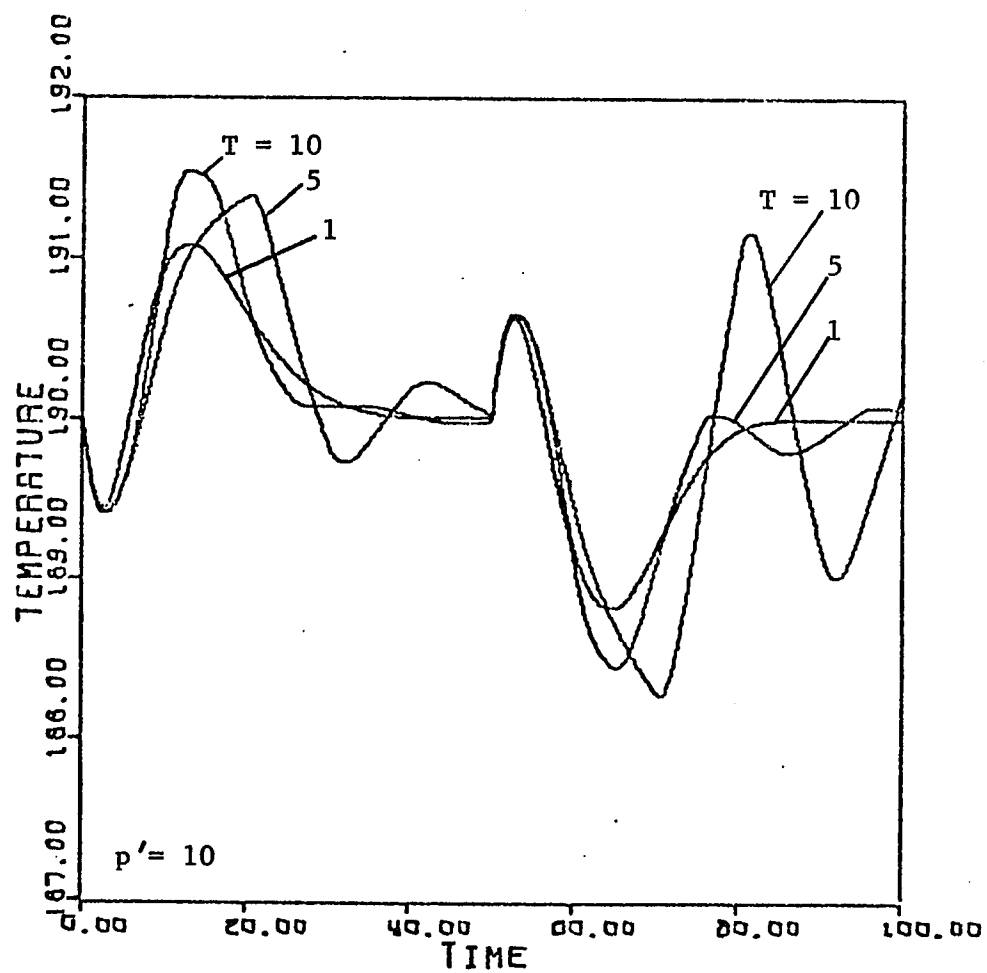


Figure 3.10c.

Responses to Load Change for
Both PID and PCID Controllers
and Sampled-data System

Summary

Optimal control theory can be applied directly to systems modeled with a first-order lag plus dead time in order to obtain tuning constants for a conventional PID analog controller. Although errors in the model and the dead time approximation will cause the system response to be suboptimal, there are several advantages due to this method. First, it is extremely easy to apply. For given model parameters, the controller constants K_c , T_i , and T_d can be obtained from generalized charts such as Figure 3.4. Another reason for using this method is that the system responses, although not optimal, appeared comparable or even better than those responses using controllers tuned with other techniques. A disadvantage could be that the parameter p in the cost function is not well defined and trouble might be encountered to determine a range of p which will produce desired results. On the other hand, the p value can be selected to suit the users needs. Larger p values will produce slower responses while smaller p values will result in tighter control. The tuning problem is reduced from a three parameter to a one parameter search.

The PCID algorithm was also tested with the tuning parameters obtained from the Riccati equation. This algorithm was found to be more successful in producing consistent results for both the load and set point problem than the conventional PID

algorithm. That is, controller parameters for a particular p value produced satisfactory responses for both the load and set point change whereas the PID had to be tuned separately for each case.

The sampled-data problem was investigated with the optimal control approach. It was found that the approximations of the sampled-data system by a continuous one, by adding a pure time delay equal to one-half the sampling time to the system dead time, was not entirely acceptable. In fact, only at fast sampling rates did the sampled-data results approach those of the continuous system, although at higher sampling times the controller could be retuned with larger p values to obtain stable results. This failure of the discrete system to produce responses comparable to those of the continuous system can be explained by the fact that in the optimal control formulation, continuous feedback of all state variables is required. Since both the discretizing of the control and approximation errors violate those conditions, it could be expected that this approach might break down.

Nomenclature

\underline{A}	State variable matrix
\underline{B}	Control matrix
c	System output
d	Disturbance or load change
e	Error, difference between output and set point
G_c	Controller transfer function
G_p	Plant or process transfer function
J	Cost function or index of performance
K	System gain
K_c	Control gain
P	Control weighting parameter in cost function
r	Set point, desired value of output
s	Differential operator
t	Time
T	Sampling time
T_d	Rate time or derivative time
T_i	Reset time or integral time
u	Control or system input
x	State vector
τ	System time constant
θ	System dead time

Literature Cited

1. Athans, Michael, "On The Design of PID Controllers Using Optimal Linear Regulator Theory," Paper ESL-P-432, Massachusetts Institute of Technology, July 1970.
2. Cohen, G. H., and G. A. Coon, Transactions ASME, Vol. 64, 1942, p. 759.
3. Johnson, C. D., "Optimal Control of the Linear Regulator with Constant Disturbance," IEEE Transactions on Automatic Control, Vol. AC-13, Aug. 1968, pp. 416-421.
4. Koppel, Lowell B., Introduction to Control Theory with Applications to Process Control, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968.
5. Lopez, A. M., P. W. Murrill, and C. L. Smith, "Tuning PI and PID Controllers," Instruments and Control Systems, February 1969.
6. Miller, J. A., A. M. Lopez, P. W. Murrill, and C. L. Smith, "A Comparison of Open-Loop Techniques for Tuning Controllers," Control Engineering, Vol. 14, No. 12, December 1967, p. 72.
7. Miller, J. A., Ph.D. Dissertation, Louisiana State University, 1970.
8. Moore, C. F., C. L. Smith, and P. W. Murrill, "Simplifying Digital Control Dynamics for Controller Tuning and Hardware Lag Effects," Instruments Practice, January 1969.
9. Rovira, A., Ph.D. Dissertation, Louisiana State University.
10. Schultz, D. G., and J. L. Melsa, State Functions and Linear Control Systems, McGraw-Hill Book Co., New York, 1967.
11. Smith, O. J. M., "Close Control of Loops with Dead Time," Chemical Engineering Progress, May 1957.
12. Van der Grinten, PMEM, "Finding Optimal Controller Settings," Control Engineering, December, 1963, p. 51.

13. Wilde, D. J. and C. S. Beightler, Foundations of Optimization, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968.
14. Ziegler, J. G., and N. B. Nichols, "Optimum Settings for Automatic Controllers," Transactions ASME, November, 1942, p. 759.

Chapter 4

MULTIVARIABLE PROCESS CONTROL

In the design and implementation of control systems for the process industries, many times a single-input, single-output system is assumed or implied. Extensive research has produced techniques for designing control loops in which one independent variable is manipulated in order to control one dependent variable. In reality however, very few examples of this type of system exist, with most processes having two or more inputs and outputs. There has until recently been relatively little theory developed for the design of control configurations for the multivariable cases because of the interaction among the individual loops. In fact, there was really no criterion for choosing which manipulated variables were to be used in controlling which control variables. The correct pairing of controlled and manipulated variables is important in order to obtain a desired or even a stable process response.

In this chapter a method of designing control loops by properly pairing the control and manipulated variables using Bristol's (1) measure of interaction is presented. The procedure is investigated by means of CHESS, a steady-state chemical process simulation program (5). Also the interaction

index as defined by Nisenfield and Schultz (7) is calculated for an example problem. Finally, the method is applied to coupled first-order lag plus dead time systems in order to illustrate the effects of dynamics on the design procedure.

Multivariable Systems

Most systems in the process industries that are to be controlled can be represented by the system in Figure 4.1. It is desired that each of the controlled variables be maintained within a certain value and to accomplish this the independent variables have to be manipulated by either an operator or an automatic controller. Depending on the nature of the process, there usually exists interaction among the independent and dependent variables. In other words, there is no one-to-one correspondence between any independent and dependent variable and a change in any of the manipulated variables will cause changes in all of the controlled variables. In an ideal situation all variables would be "decoupled"; that is, each manipulated variable affects only one controlled variable. Then the system could be reduced to a number of single-input, single-output loops. However, most systems do contain some degree of interaction and this interaction is a major factor in designing the control system. Consider the following two input-two output process and its control scheme (Figure 4.2).

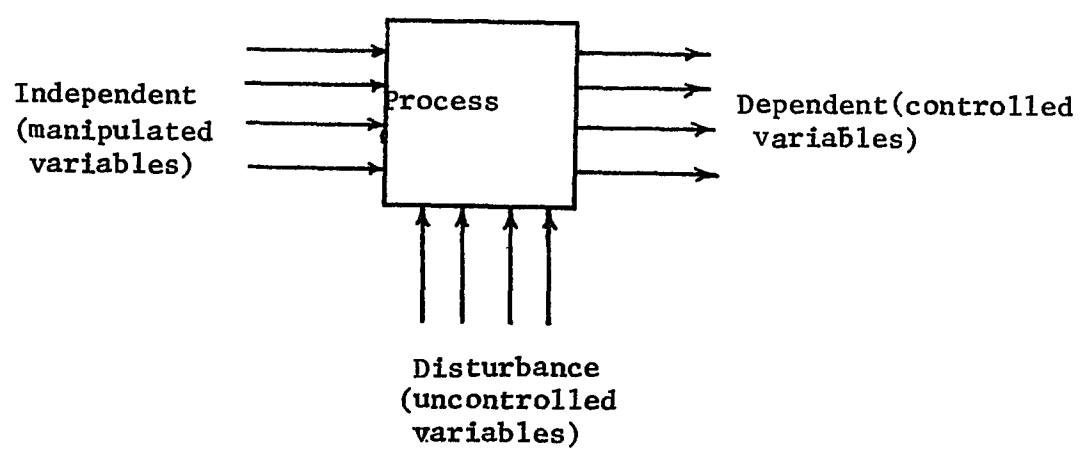


Figure 4.1. Typical multivariable system

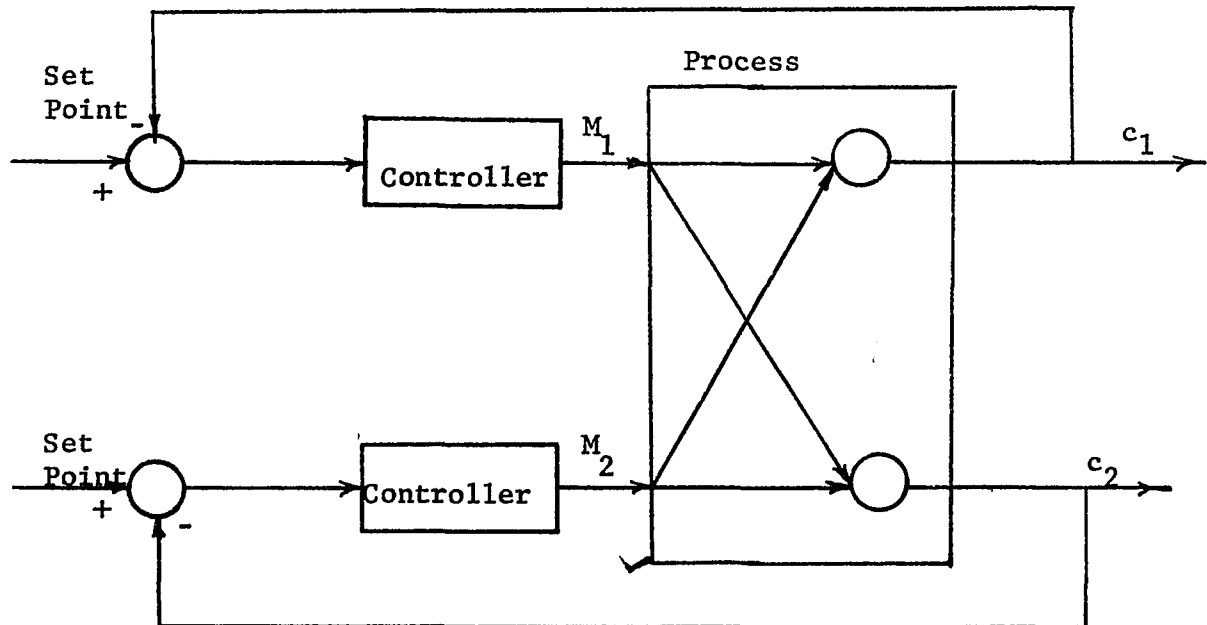


Figure 4.2. Closed loop control of system with two inputs/
two outputs.

The diagonal arrows in the process represent the interaction terms and if they were negligible, two decoupled loops would exist. In the design of the above system, some criteria had to be applied in order to determine that the manipulated variable M_1 would be used to control C_1 and that M_2 would regulate C_2 . In some applications an operator who is familiar with the process might be able to pair the variables satisfactorily on the basis of past experience, but with a new or complex process this may not be possible. Thus, a technique was needed in order to dictate to the designer how to pair the variables for multivariable systems.

The method that will be presented here is based on the relative process gain matrix as discussed by Shinskey (9). The development of the theory behind this matrix may be explained by two similar lines of thought. Shinskey (9) defines the measure of interaction for multivariable processes in which the elements of the matrix are a ratio of open loop to closed loop steady state gains. This measure is really based on the design technique presented by Bristol (1). Bristol states that since all elements have to be compared to each other, they have to be normalized in order to remove their dependency on their particular units. The method used to normalize the open loop matrix results in the same matrix as defined by Shinskey. Although the reasoning behind the development of this matrix is slightly different, the actual determination of the numerical values of the elements in the matrix will be exactly the same. From the values of the elements

an indication of the relationship or interaction among the various variables can be observed. This will make it possible for someone to design a control scheme with the best possible arrangement or pairing of variables. Nisenfield and Schultz (7) use the elements of the relative gain matrix in their definition of the interaction index which is an indication of whether a particular variable pairing will produce stable or unstable responses. For the worst possible dynamic case, an interaction index of less than unity (absolute value) denotes a stable configuration.

Relative Process Gain

In Bristol's development, the elements of the relative process gain matrix are defined as

$$\lambda_{ij} = \frac{\partial C_i / \partial M_j \big|_M}{\partial C_i / \partial M_j \big|_C} \quad (4.1)$$

where

- λ_{ij} = relative process gains
- C_i = controlled variables
- M_j = manipulated variables

The numerator or open loop gain is a change in an output variable when one of the input variables is changed and the other manipulated variables are held constant. The denominator or closed loop gain is a change in one of the controlled variables divided

by the corresponding change in an input variable with the remainder of the controlled variables held constant. Thus, the definition represents the open loop response divided by the closed loop response. Another important property of this matrix is that the expression is dimensionless and normalized with the sum of any row or column equal to one. This characteristic makes it unnecessary to calculate every element in the matrix or can be a helpful check to see if the elements were determined correctly. The feature of being dimensionless indicates that all elements are comparable and inferences of their relative sizes can be made. Other properties of the measure can be found by referring to the original work of Bristol.

The sizes of the elements of the relative gain matrix suggest to the designer several facts about the process investigated. First, larger values denote stronger dependency of a controlled variable due to that particular manipulated variable. This is the basis of the design technique, that variables with the largest positive relative gains should be paired for closed loop control. This will yield the configuration that maximizes the probability of being able to obtain good control. A relative gain approaching zero indicates a controlled variable that is insensitive to any change in the corresponding manipulated variable. Systems in which there exist one or more loops that can be decoupled will have a relative gain matrix containing "ones" and "zeroes".

In some cases a designer might be unaware that such a loop exists if the process contains many possible combinations of variables and no previous information is available. Also brought to light are combinations that produce unstable responses and are uncontrollable regardless of controller settings. These pairings are represented by negative relative gains and must be avoided by the controller designer. Other inferences that can be made with the matrix are the values for controller settings. Tuning controllers for interacting loops is quite difficult and Shinskey suggests a method of using the relative gains to determine controller settings which will be discussed in the section concerning dynamics.

An example to illustrate the mechanics of determining the relative gain matrix is the blending system shown in Figure 4.3. Two streams with flow rates Y and Z lbs/hr are mixed to form a product stream of composition x and flow rate F . The two independent or manipulated variables, Y and Z are to control the two dependent variables F and x . It must be decided to which valve the flow and composition controllers will be connected. The equations for the above process are:

$$\text{Overall material balance: } F = Y + Z \quad (4.2)$$

Material balance on

$$\text{component Y: } x = \frac{Y}{F} \quad (4.3)$$

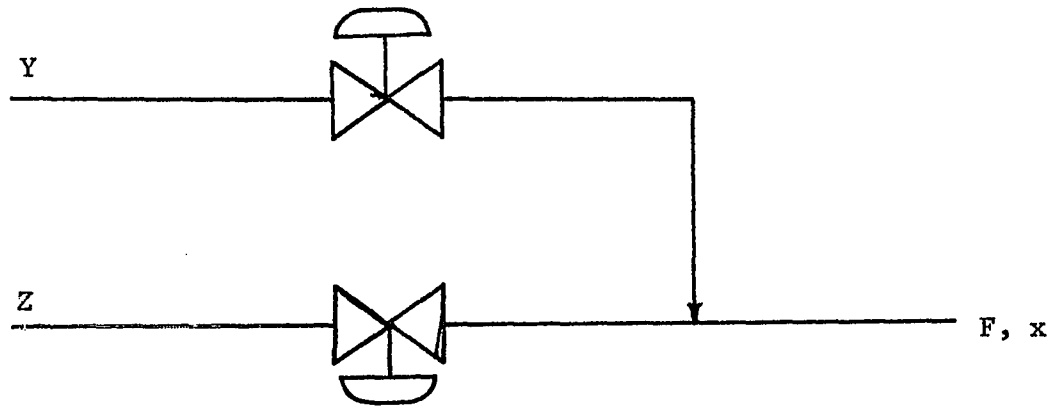


Figure 4.3. Diagram for Blending System.

From these two equations, the elements of the relative gain matrix are calculated. From the definition of the matrix equation 4.1 will look like:

$$\begin{array}{cc}
 & \begin{array}{cc} Y & Z \end{array} \\
 \begin{array}{c} F \\ \\ x \end{array} & \left[\begin{array}{cc}
 \frac{\partial F}{\partial Y} \Big|_Z & \frac{\partial F}{\partial Z} \Big|_Y \\
 \frac{\partial F}{\partial Y} \Big|_x & \frac{\partial F}{\partial Z} \Big|_x \\
 \frac{\partial x}{\partial Y} \Big|_Z & \frac{\partial x}{\partial Z} \Big|_Y \\
 \frac{\partial x}{\partial Y} \Big|_F & \frac{\partial x}{\partial Z} \Big|_F
 \end{array} \right]
 \end{array} \quad (4.4)$$

From the system equations

$$F = Y + Z \Rightarrow \frac{\partial F}{\partial Y} \Big|_Z = 1 \quad \frac{\partial F}{\partial Z} \Big|_Y = 1$$

$$x = \frac{Y}{Z+Y} \Rightarrow \frac{\partial x}{\partial Y} \Big|_Z = \frac{Z}{(Z+Y)^2} = \frac{Z}{F^2} = \frac{1-x}{F}$$

$$\frac{\partial x}{\partial Z} \Big|_Y = -\frac{Y}{(Z+Y)^2} = -\frac{Y}{F^2} = -\frac{x}{F}$$

By rearrangement

$$xF = Y \Rightarrow \left. \frac{\partial F}{\partial Y} \right|_x = \frac{1}{x} \qquad \left. \frac{\partial x}{\partial Y} \right|_F = \frac{1}{F}$$

$$F = \frac{Z}{1-x} \Rightarrow \left. \frac{\partial F}{\partial Z} \right|_x = \frac{1}{1-x} \qquad \left. \frac{\partial x}{\partial Z} \right|_F = -\frac{1}{F}$$

Substituting into equation 4.4 yields the relative gain matrix for this system.

$$\begin{array}{cc|cc} & & Y & Z & \\ & & \hline F & & x & 1-x & \\ x & & 1-x & x & \end{array} \quad (4.5)$$

As can be seen the value of the matrix depends on the operating level of the process. For small concentrations of Y, the flow rate F is more responsive to changes in Z while the overall concentration depends largely on Y. For concentrations greater than 0.5 the reverse is true and when x equals 0.5, F and x are equally affected by both the manipulated variables. The reason for this dependency on operating level is that the process equations are nonlinear resulting in a relative gain matrix that is not constant and is a function of the system parameters. This fact will become important later when the derivatives will be changed to

difference equations. The perturbations for these difference equations must be kept as small as possible so that the definitions will still hold even though a non-linear process is being investigated. Also the results above display the properties that the relative gain matrix is dimensionless and the sum of any row or column is equal to one.

In the above example the process equations were available and the relative gain matrix was analytically calculated directly from its definition, which was in terms of partial derivatives. However, many times these equations are not known and a physical system has to be investigated experimentally or by analog or digital simulation. The following difference form of the definition could then be applied.

$$\lambda_{ij} = \frac{\left. \frac{\Delta C_i}{\Delta M_j} \right|_M}{\left. \frac{\Delta C_i}{\Delta M_j} \right|_C} \quad (4.6)$$

When studying these systems the open loop or steady state gains of the manipulated variables are relatively easy to calculate. The controllers are placed on manual and each of the manipulated variables are perturbed. Their effect on each of the controlled variables can then be recorded and the gains obtained from the

numerator of equation 4.6. An important point to remember is that small changes must be made in order to eliminate or reduce the effect of nonlinear terms of the process. This technique is essentially a linearization procedure around an operating point and as the perturbations approach zero, the difference equations 4.6 approach those for the derivatives (equation 4.1.). The closed loop gains or denominators of equation 4.6 are found by a similar procedure. The controllers are placed on automatic and the set points of each of the controlled variables are perturbed. After steady state is reached, the corresponding changes in the manipulated variables are noted and the difference equation will yield the closed loop gains. Equation 4.6 is then applied with the open loop gains divided by the closed loop gains to find the relative gain matrix.

Although the above procedure is a straight forward application of the definition, the relative gain matrix would seldom be calculated in this manner. After the open loop steady state gains have been computed, Shinskey shows how a simple matrix operation will yield the relative gain matrix. If the transpose of the inverse of the steady state gain matrix is multiplied term by term with the original steady state matrix, the result will be the relative gain matrix. This may be written as

$$\lambda_{ij} = (M_{ij})(L_{ij}) \quad (4.7)$$

where

λ_{ij} are the elements of the relative gain matrix

M_{ij} are the elements of the steady state gain matrix

and

$$\underline{L} = (\underline{M}^{-1})^T$$

It is obvious that the determination of the relative gain matrix in this manner is more practical because fewer disturbances and measurements have to be made on the process. Shinskey explains that the desired matrix can also be calculated with the above technique with data from closed loop tests. However, the elements of the M matrix must be inverted and are defined as $(\Delta M_j / \Delta C_i) |^c$ instead of $(\Delta C_i / \Delta M_i) |^c$. This matrix can then be inverted and transposed and the resulting matrix multiplied term by term with the original M matrix. The product will be the relative gain matrix. Many times for actual processes the closed loop tests are run because the values of the controlled variables must remain within certain limits.

The Interaction Index

Nisenfield and Schultz make use of the elements of the relative gain matrix in their definition of the interaction index. It is given by the equation

$$I_{ij} = \frac{1 - \lambda_{ij}}{\lambda_{ij}} \quad (4.8)$$

where I_{ij} is the interaction index for the control variable i and the manipulated variable j . The λ_{ij} are the relative gains for that variable pair as given by equation 4.1. The interaction index is derived from the geometric progression the system experiences after an upset has occurred. The open and closed loop gains describe how the system attempts to compensate for the upset. These are dynamic responses of the system and the particular time constants for each loop will determine exactly how fast the interacting loops will react. The interaction index assumes the worst possible case; that is, each controller takes corrective action before the interaction effect is detected. The series of individual responses can be described by

$$\varphi_i = \epsilon I_{ij}^{(N+1)} \quad (4.9)$$

where

φ_i = value of control variable i after $N-1$ half-cycles

ϵ = magnitude of disturbance or upset

I_{ij} = interaction index of variables i, j

N = number of half-cycles

As can be seen, if the absolute value of the interaction index is less than one, the response of the process will be damped and eventually approach zero. If the interaction index equals one

a uniformly oscillating system exists and indexes greater than one denote that the upsets will be amplified until the limits of the process are reached. Consequently, the interaction index may be incorporated with Shinskey's design technique in two different ways. First, the variable pairing with the smallest absolute index will produce responses which are more stable, tending to reach steady state in the fewest number of cycles. Thus, the choice of variable pairing will be determined by the smallest interaction indexes. Also, the index can be used to predict the number of cycles it will take for the process to reach some particular level. By solving equation 4.9 for N, the number of cycles the system oscillates before it is damped to a specified value can be found. This is given by

$$N = \frac{\ln \phi - \ln \epsilon}{\ln I_{ij}} + 1 \quad (4.10)$$

where

N is the number of half cycles

ϵ is the initial disturbance

I_{ij} is the interaction index

ϕ_i is the specified level of the controlled variable

Dynamic Control

Up to now all that has been discussed has dealt with steady-state control. This is almost an idealized situation because in

real processes even though steady state will be achieved for periods of time, there will always exist system upsets. These upsets might be in the form of set point changes as imposed by the process operators or uncontrolled load changes such as changing feed rates, loss of cooling water capacity due to ambient temperature conditions, changes in feed composition, or even loss of equipment efficiency. These factors appearing either over a long period of time or as instantaneous step functions will cause changes in process operating conditions. The control system must compensate for these changes and bring the system's controlled variables back to the desired levels. During these periods of time, a dynamic control problem will exist and it will be necessary to determine how the relative gain matrix can be used to predict responses for these situations.

There are really two cases that appear in the investigation of dynamic control. The first is when all the elements of the relative gain matrix are between the values of 0 and 1. Shinskey suggests that this case is produced for systems that can be made stable for both steady-state and dynamic control, such as those designed only with material balance control strategy. Advanced control techniques such as variable decoupling and multivariable controller tuning can be applied to these systems. The second case arises when some of the elements of the relative gain matrix are greater than 1 and the remainder are negative.

For these cases, the matrix indicates that both steady state and dynamic control are impossible and that advanced control techniques cannot be applied directly. This case will be discussed for a general first-order lag plus dead time model.

For simple control systems, in order to determine values for controller constants a step function can be applied to the independent variable and its effect on the dependent or controlled variable can be established. However, in multivariable systems, each independent variable usually affects each dependent variable so that controller tuning becomes very difficult. To remedy this situation it has been proposed to use decouplers in order to force one input variable to affect only one output variable. Mathematically, coupled and decoupled systems can be written as:

$$\underline{C} = \underline{K} \underline{M} \quad (4.11)$$

where \underline{C} is the dependent variable vector of size i

\underline{M} is the independent variable vector of size j

\underline{K} is an $i \times j$ matrix

If the system is linear, the elements of \underline{K} will be constants and if not, they will be of a more complex form. For decoupled systems only the diagonal terms of the \underline{K} matrix are non-zero whereas

for coupled systems, any element of \underline{K} may be non-zero. Thus, the thought behind a decoupler is to manipulate the system equations so that each output will be dependent on only one input. In reality all that is being done in the decoupler is reversing the process equations which forces the decoupler equations to be of the same form mathematically as the process equations. For this reason, if a complex process is being investigated, the decoupler will also be complex. An example of how to obtain a steady-state decoupler will be presented later.

Tuning relationships are well established for single-input, single-output systems. However, quite a different picture exists for multivariable processes. For most processes that consist of two loops, one that reacts fast and the other that responds slow, the approach is to tune the fast loop as tight as possible and the slow loop rather loose. Tuning is performed on one loop while the other is placed in the manual position. The theory or reasoning behind this approach is that the fast loop is affected very little by the coupling and can be controlled very close. The slow loop on the other hand is more sensitive to changes and cannot be regulated closely. Shinskey presents a tuning technique for multivariable systems using the relative gain matrix, somewhat compensating for variable interaction. Each loop is tuned separately as described previously. Then the controller gains are reduced by multiplying each by the

corresponding element of the relative gain matrix. Since the relative gain matrix is defined as the ratio of open loop to closed loop gains, this procedure might appear reasonable. However, the technique breaks down when values for the relative gain matrix are greater than unity. As explained previously, relative gains greater than one are accompanied by negative relative gains indicating dynamic instability. In order to control a system of this form, care must be taken to reduce controller gains (loosen control) in order to produce stable responses.

Multivariable Control of an Absorber-Stripper Process -

Steady State Case

A systematic analysis of all variables must be made in order to properly design a control system for complex multivariable processes. All possibilities of controlled and manipulated variables should be listed along with any disturbances that might enter into the control loops. Then a study to determine the effects of each of the manipulated variables upon each of the controlled variables by means of a simulation program or by experiments upon the actual system has to follow. These effects along with the variable interactions will next be interpreted utilizing the measure known as the relative gain matrix. From the values of this matrix the designer is able to correctly pair manipulated and controlled variables. Also, the relative gain

matrix is used to calculate an interaction index for each pair of variables from which an indication of the stability of the loop will be obtained.

The process investigated here is an absorption-stripping operation in which methane is removed from natural gas by the absorption of the heavier components of the natural gas into a lean oil. The lean oil is regenerated in a distillation column, the lighter gases recovered from the column overhead and the lean oil being the bottoms product. The entire configuration of the process along with the conditions of the feed streams is given in Figure 4.4 and Table IV-1.

This system will be studied by means of CHESS, a Chemical Engineering System Simulation program. The program is composed of separate modules, each representing a different unit operation. The user specifies to the program how these modules should be arranged by connecting them with process streams. The input streams must be specified with all necessary stream conditions. CHESS will then solve for intermediate and product streams using the steady state material and energy balances within each process module. Since most chemical processes contain recycle loops that are unknown or dependent on other variables, the user will be unable to specify these streams. In these cases, the program employs an iterative scheme, assuming unknown values and iterating until all loops converge.

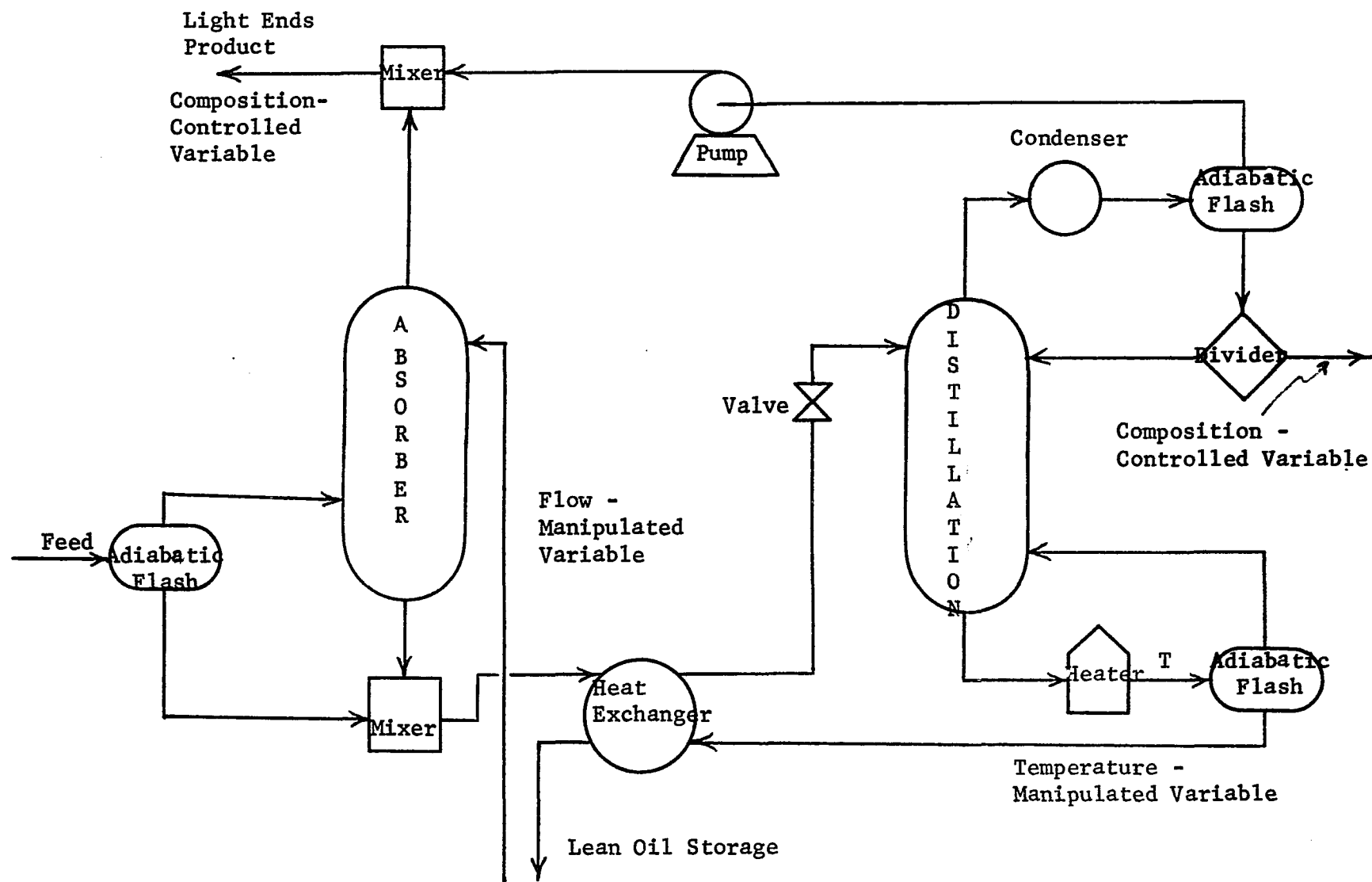


Figure 4.4. Absorber-stripper natural gas process.

Table IV-1
Stream Conditions for
Absorption-Stripper Process

Gas Feed Stream:

Methane	24.0 lb-moles/hr
Ethane	1.0
Propane	1.0
I-Butane	0.6
N-Hexane	<u>0.4</u>
	27.0 lb-moles/hr
Temperature	= 550°R
Pressure	= 300 PSIA

Lean Oil Feed:

N-Tetradecane	18.0 lb-moles/hr
Temperature	= 550°R
Pressure	= 300 PSIA

For this study the variables to be controlled will be the absorber overhead and distillation column overhead. The flow rate of lean oil and reboiler temperature are chosen to be the manipulated variables. The CHES program allows the user to specify both oil flow rate and reboiler temperature along with other independent variables in the problem which will be considered constant. The proper pairing of these variables for control will be determined from the values of the relative gain matrix. The matrix will be computed from the open loop gains M from which matrix manipulation will yield the design matrix. In order to obtain the open loop gains three computer runs have to be made. The first will be the base run and the other two runs will have each of the manipulated variables perturbed. Thus, the effect of each independent variable is measured and the steady state gains calculated. The above procedure was followed and the results obtained by changing the lean oil flow rate from 18 to 20 lb-moles/hr and the reboiler temperature from 800°R to 900°R are in Table IV-2. Values for the concentration are based on the mole fraction of the heavy oil. The steady state gains defined as $\Delta C / \Delta M|_M$ can be easily calculated from the above results. If the concentration of lean oil in the absorber overhead is defined as x_1 and that in the overhead product of the distillation column as x_2 the steady state gains may be written as:

Table IV-2
Steady State Concentrations of the
Controlled Variables

	<u>Absorber OVHD</u>	<u>Distillation OVHD</u>
	x_1	x_2
<u>Base Run</u>	.2005(10 ⁻⁴)	.435
Lean Oil = 20 lb/moles	.2010(10 ⁻⁴)	.476
Reboiler T = 900°R	.198 (10 ⁻⁴)	.212

	Flow	Temperature
x_1	$\left. \frac{\Delta x_1}{\Delta F} \right _T$	$\left. \frac{\Delta x_1}{\Delta T} \right _F$
x_2	$\left. \frac{\Delta x_2}{\Delta F} \right _T$	$\left. \frac{\Delta x_2}{\Delta T} \right _F$

(4.12)

The elements are found

$$\left. \frac{\Delta x_1}{\Delta F} \right|_T = \frac{.201(10^{-4}) - .2005(10^{-4})}{2} = 2.5(10)^{-8}$$

$$\left. \frac{\Delta x_2}{\Delta F} \right|_T = \frac{.476 - .435}{2} = 2.05(10)^{-2}$$

$$\left. \frac{\Delta x_1}{\Delta T} \right|_F = \frac{.198(10)^{-4} - .2005(10)^{-4}}{100} = -2.5(10)^{-9}$$

$$\left. \frac{\Delta x_2}{\Delta T} \right|_F = \frac{.212 - .435}{100} = -2.23(10)^{-3}$$

or

$$\underline{M} = \begin{vmatrix} 2.5(10)^{-8} & -2.5(10)^{-9} \\ 2.05(10)^{-2} & -2.23(10)^{-3} \end{vmatrix}$$

From equation 4.7 the relative gain matrix is found by inverting and transposing \underline{M}

$$(\underline{M}^{-1})^T = \frac{1}{(+51.25 - 55.75)10^{-12}} \begin{vmatrix} -2.23(10)^{-3} & -2.05(10)^{-2} \\ 2.5(10)^{-9} & 2.5(10)^{-8} \end{vmatrix}$$

The relative gain matrix from 4.7 is

$$\lambda_{ij} = M_{ij} * (M_{ij}^{-1})^T$$

$$\underline{\lambda} = - \frac{1}{4.5(10)^{-12}} \begin{vmatrix} -5.575(10)^{-11} & 5.125(10)^{-11} \\ 5.125(10)^{-11} & -5.575(10)^{-11} \end{vmatrix}$$

$$\underline{\lambda} = \begin{vmatrix} 12.4 & -11.4 \\ -11.4 & 12.4 \end{vmatrix}$$

or

	Flow	Temperature
x_1	12.4	-11.4
x_2	-11.4	12.4

From these values and the design criteria, one should pair the absorber overhead composition with the lean oil flow and the distillation column overhead composition with the reboiler

temperature. This combination would probably have been decided on anyway but the relative gain matrix settles any doubt. The negative signs in the matrix suggest that the pairing of lean oil flow and absorber overheads will form a system that is impossible to control. This fact is important in cases where the variable pairing is not obvious.

It might also be helpful to calculate the interaction index for this system. From equation 4.8 the interaction index is given by

$$I_{ij} = \frac{1 - \lambda_{ij}}{\lambda_{ij}}$$

Since this is a two-by-two system, there will be two possible combinations of pairing the variables. The index will indicate which combination is the easier to control. The criteria is that if the absolute value of the interaction index is less than one the system will be stable and controllable. If greater than one the system will oscillate until its physical limits are reached. For this system the correct pairing has an index of

$$I_{x_1 F} = \frac{1 - 12.4}{12.4} = - .92$$

and the other possibility has an index of

$$I_{x_2 F} = \frac{1 + 11.4}{-11.4} = - 1.09$$

These values suggest that the process is difficult to control with its response being oscillatory. Also, the choice of pairing that was decided on by the relative gain matrix is the same as indicated by this method.

Multivariable Control of a Simple Model - Dynamic Case

In order to investigate the use of the relative gain matrix for predicting dynamic behavior, some dynamic terms must be incorporated into a mathematical model. CHESS, being a steady-state simulator cannot be applied in dynamic studies except to calculate open loop steady state gains. Because the absorber-stripper process is known to be non-linear and difficult to model accurately, a simple first-order lag plus dead time model will be used to test the dynamic case. Even though the steady state gains obtained in the preceding section will be used in the transfer functions for the dynamic model, the responses are not suggested to represent those of the absorber-stripper process. The following transfer functions were used to study the dynamic case:

$$Y_A(s) = G_{AF}(s)F(s) + G_{AT}(s)T(s) \quad (4.13)$$

$$Y_D(s) = G_{DF}(s)F(s) + G_{DT}(s)T(s) \quad (4.14)$$

where

$Y_A(s)$ and $Y_D(s)$ are the controlled variables

$F(s)$ and $T(s)$ are the manipulated variables

$G_{ij}(s)$ are the model transfer functions

Each model is of the form

$$G(s) = \frac{Ke^{-st_0}}{\tau s + 1} \quad (4.15)$$

where

K is the process gain

τ is the process time constant

t_0 is the process dead time

The process gains can be determined by the relationship

$$K_{AF} = \frac{\partial Y_A}{\partial F} = \frac{Y_A(t_f) - Y_A(0)}{F(t_f) - F(0)} \quad (4.16)$$

These gains can be seen to be the same as the open loop gains that were previously calculated and used in the determination of the relative gain for the absorber-stripper process. Thus, they can be applied directly to the process model. The values for the time constants and dead times were assigned arbitrary values, and therefore may not be indicative of the performance of the absorber-stripper. However, they will illustrate the importance of dynamic considerations. The values used appear in Figure 4.5 and Table IV-3. Substituting the gains, time constants, and dead times into the model, the following system was used to study the dynamic responses:

$$Y_A(s) = \frac{2.5(10)^{-8} e^{-.1s}}{2s + 1} F - \frac{2.5(10)^{-9} e^{-.2s}}{4s + 1} T \quad (4.17)$$

$$Y_D(s) = \frac{2.05(10)^{-2} e^{-.1s}}{2s + 1} F - \frac{2.23(10)^{-3} e^{-.2s}}{4s + 1} T \quad (4.18)$$

The simplest test was to determine whether a conventional PI controller on each loop could be found that would satisfactorily control the system. Two cases were proposed as to the pairing of variables. The tuning parameters for the controllers were found using the relationships suggested by Lopez(6).

For the ITAE criterion function these relationships are given:

$$KK_c = .859 (t_o/\tau)^{-.977} \quad (4.19)$$

$$\tau/T_i = .674(t_o/\tau)^{-.68} \quad (4.20)$$

where

K is the system gain

τ is the system time constant

t_o is the system dead time

K_c is the controller gain

T_i is the controller reset time

Table IV-3

Model Parameters for Multivariable Study

General Equation:

$$\frac{C_i}{M_j} = \frac{K_{ij} e^{-t_{ij}s}}{\tau_{ij}s+1}$$

$$C_i = Y_A$$

$$C_i = Y_D$$

$$M_j = F$$

$$K_{AF} = 2.5E-08$$

$$K_{DF} = 2.05E-02$$

$$\tau_{AF} = 2.0$$

$$\tau_{DF} = 2.0$$

$$t_{AF} = 0.1$$

$$t_{AF} = 0.1$$

$$M_j = T$$

$$K_{AT} = -2.5E-09$$

$$K_{DT} = -2.23E-03$$

$$\tau_{AT} = 4.0$$

$$\tau_{DT} = 4.0$$

$$t_{AT} = 0.2$$

$$t_{DT} = 0.2$$

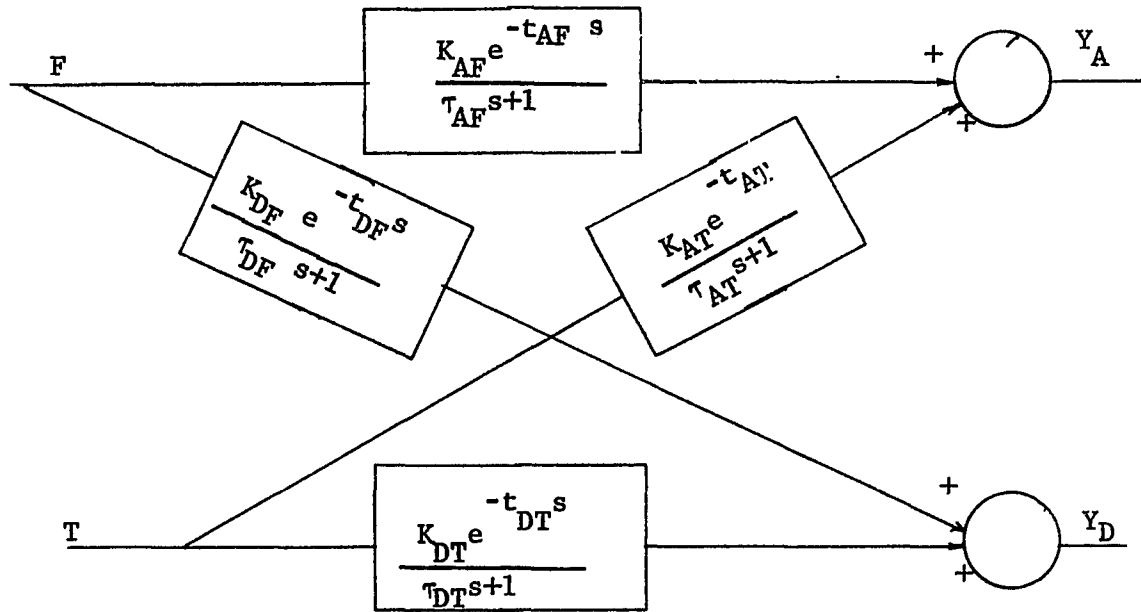


Figure 4.5. Model diagram for multivariable study.

For the first case the single loop control was implemented; that is, the Y_A variable was controlled by manipulating F . The response seen in Figure 4.6 shows that for a set point change Y_A can be satisfactorily controlled, but Y_D drifts toward a new steady state. Similarly, Y_D regulated by changing T produces an analogous response as shown in Figure 4.7. Finally, both loops are closed with the variable pairing $Y_A - F$ and $Y_D - T$ as dictated by the relative gain matrix in the preceding section. The same set of controller constants produced an unstable system as can be seen in Figure 4.8. Thus, the multivariable system cannot be controlled by tuning each loop individually. In order to verify that the negative signs in the relative gain matrix correctly predicted uncontrollable pairing, the pairings were switched. As expected, an unstable solution was obtained for all controller settings when both the loops were closed. This was caused by the action of the manipulated variables F and T opposing the effect of each other and resulting in instability.

In an attempt to produce more effective control, the steady state decoupler as suggested by Shinskey was incorporated into the system. The decoupler is calculated from the gains of the process model and is shown in Figure 4.9. An inspection of the values of the decoupler gains show that they are really the elements of the inverse of the open loop steady state gain matrix. From the relationship

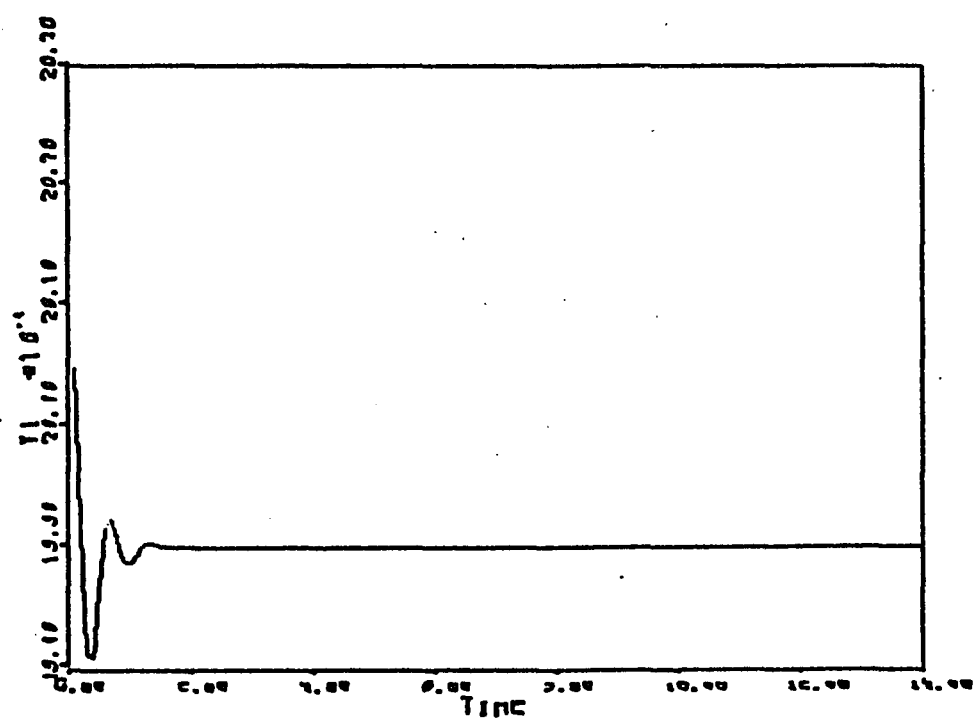
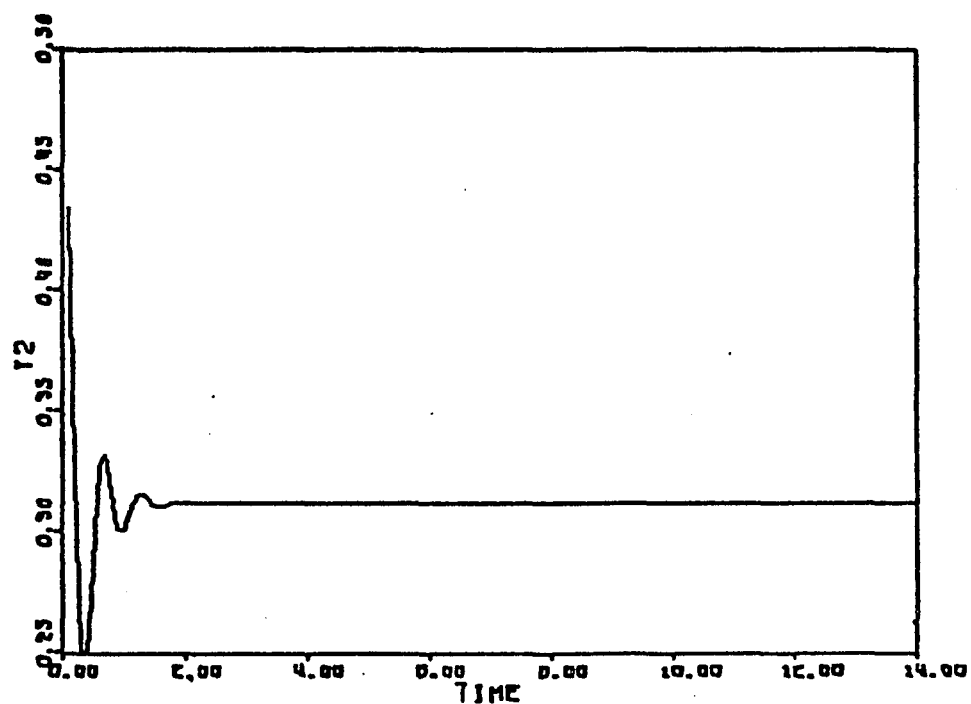


Figure 4.6. Set point response for closed loop on Y1-F.

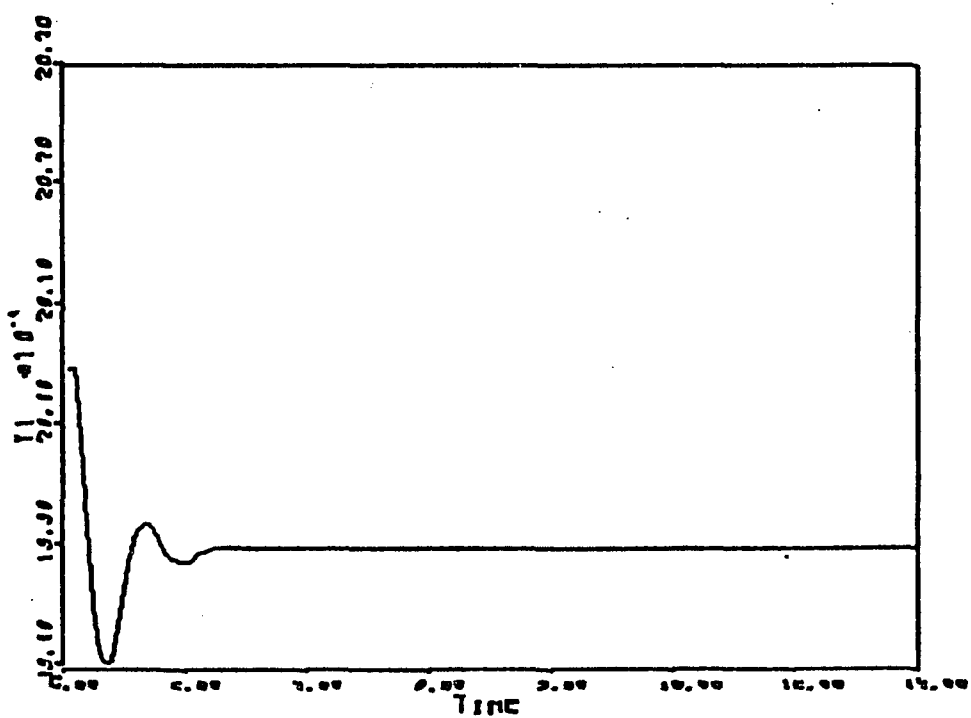
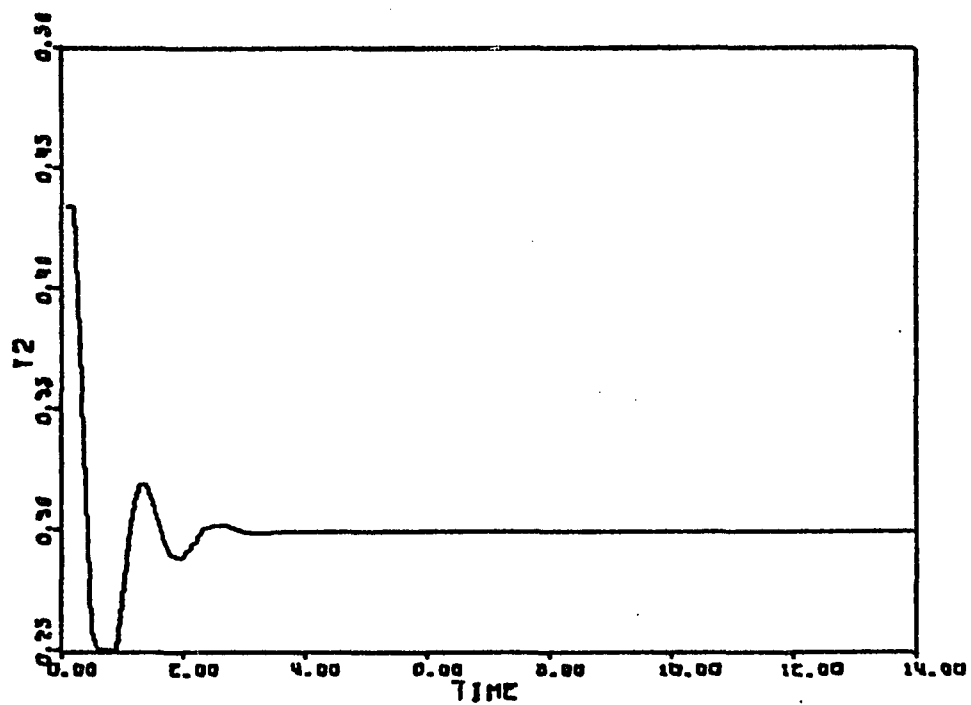


Figure 4.7. Set point responses for closed loop on Y2-T.

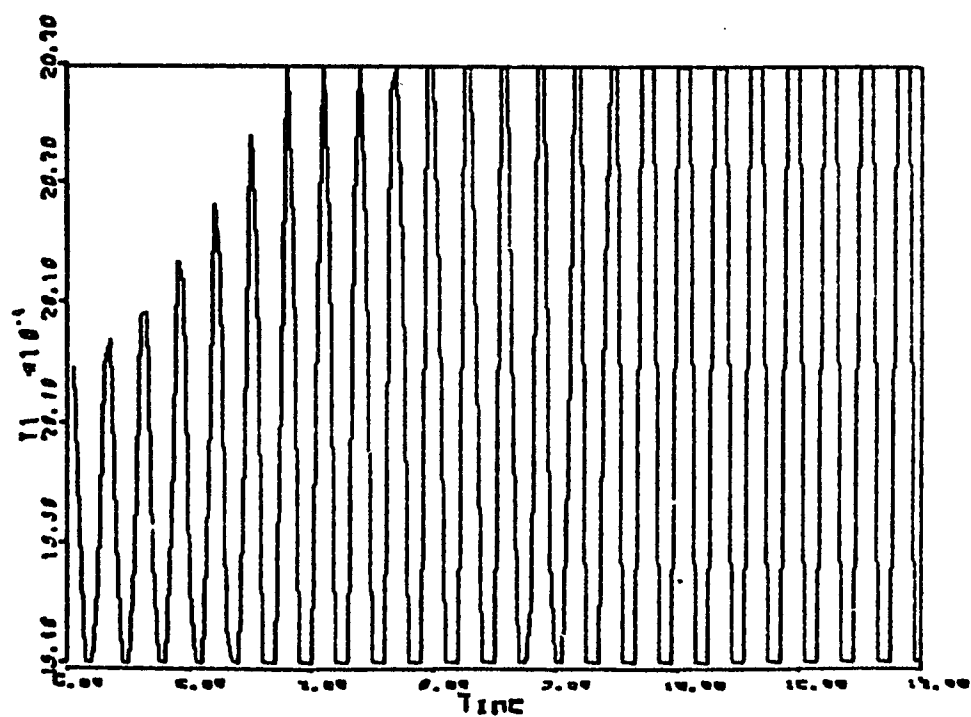
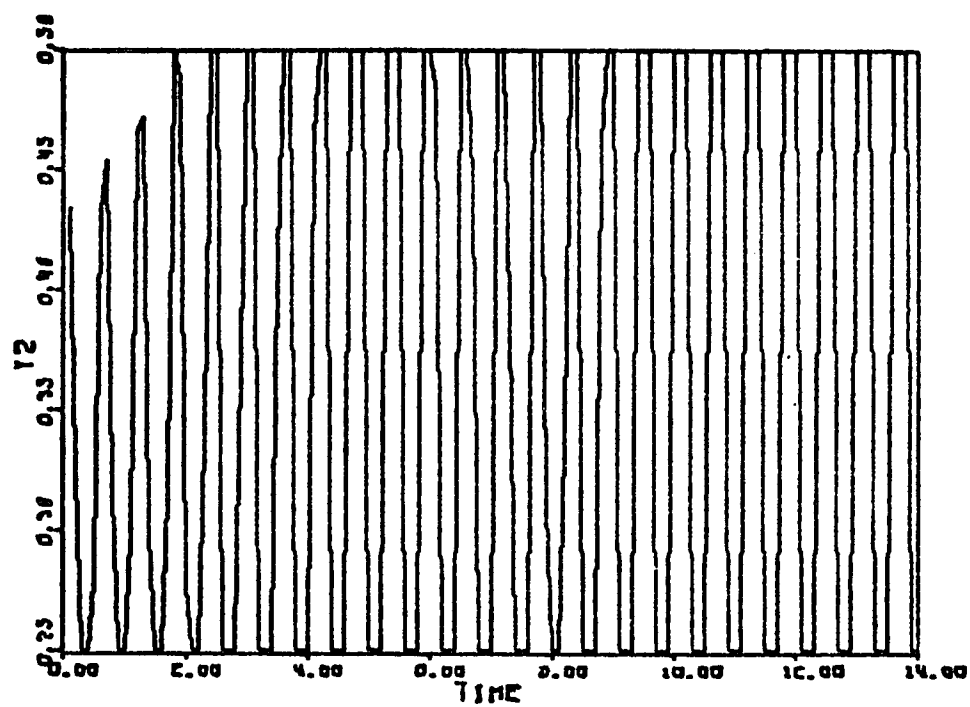


Figure 4.8. Set point response for both loops closed.

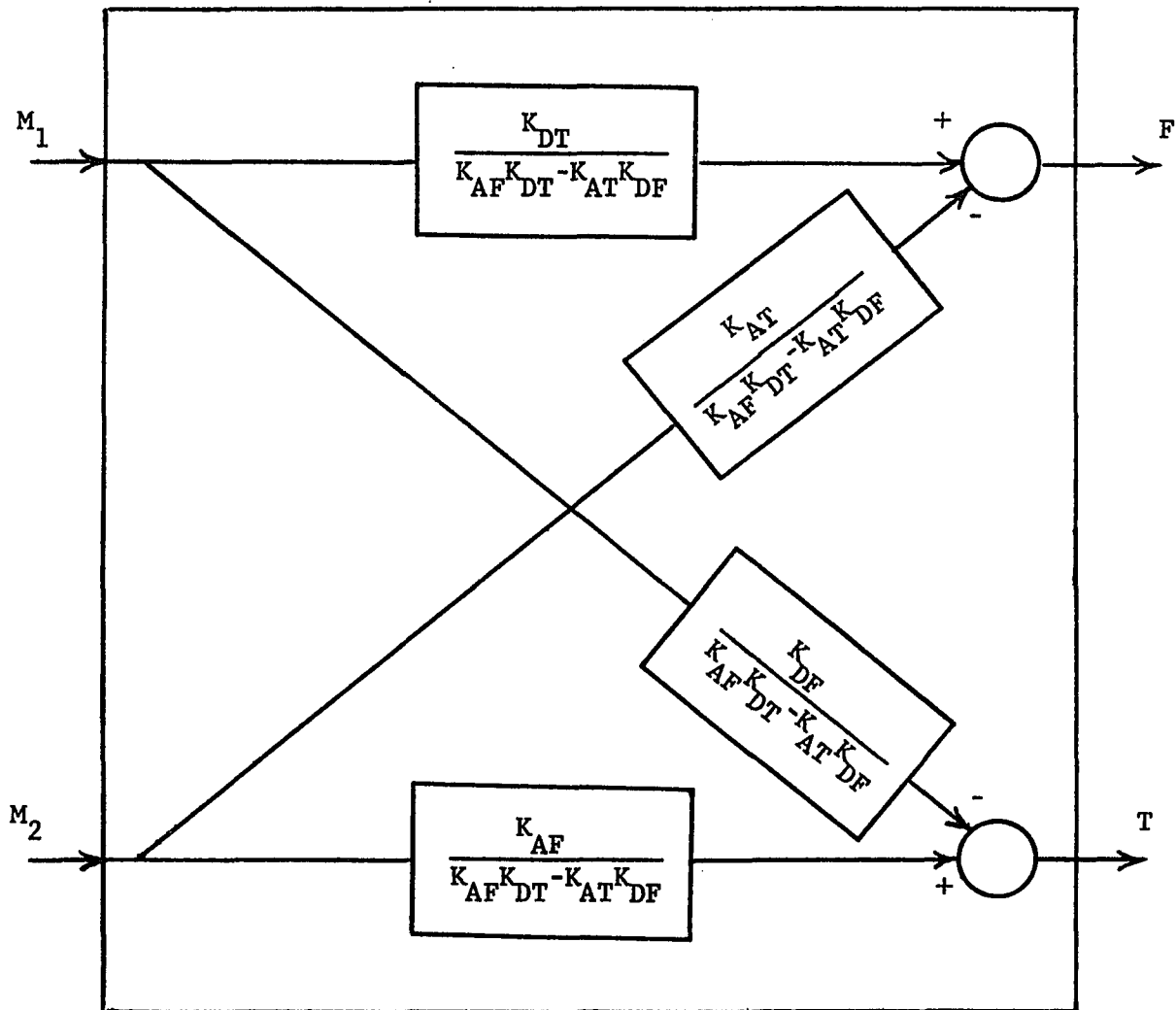


Figure 4.9. Steady State Decoupler.

$$\begin{bmatrix} Y_A \\ Y_D \end{bmatrix} = \begin{bmatrix} K \end{bmatrix} \begin{bmatrix} F \\ T \end{bmatrix} \quad (4.21)$$

it is desired to obtain the form

$$\begin{bmatrix} Y_A \\ Y_D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (4.22)$$

Since M_1 is directly related to F and M_2 is related to T (from the controller pairings), then the following must hold:

$$\begin{bmatrix} F \\ T \end{bmatrix} = \begin{bmatrix} K \end{bmatrix}^{-1} \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (4.23)$$

However, the assumption made was that \underline{K} is composed only of the steady state gains. This is incorrect because the system is time dependent; thus, the steady state decoupler does not take into account the effects of dynamics. The problem that arises can be seen from the responses of the steady state decoupled system due to step inputs. Figure 4.10 and 4.11 are the results of perturbing each of the inputs into the decoupler, that is, M_1 and M_2 . Remembering that the decoupler forces the open loop gains to unity, notice how the responses change significantly before returning to their steady state values. In fact, the response of Y_D due to a change in M_2 actually results in the

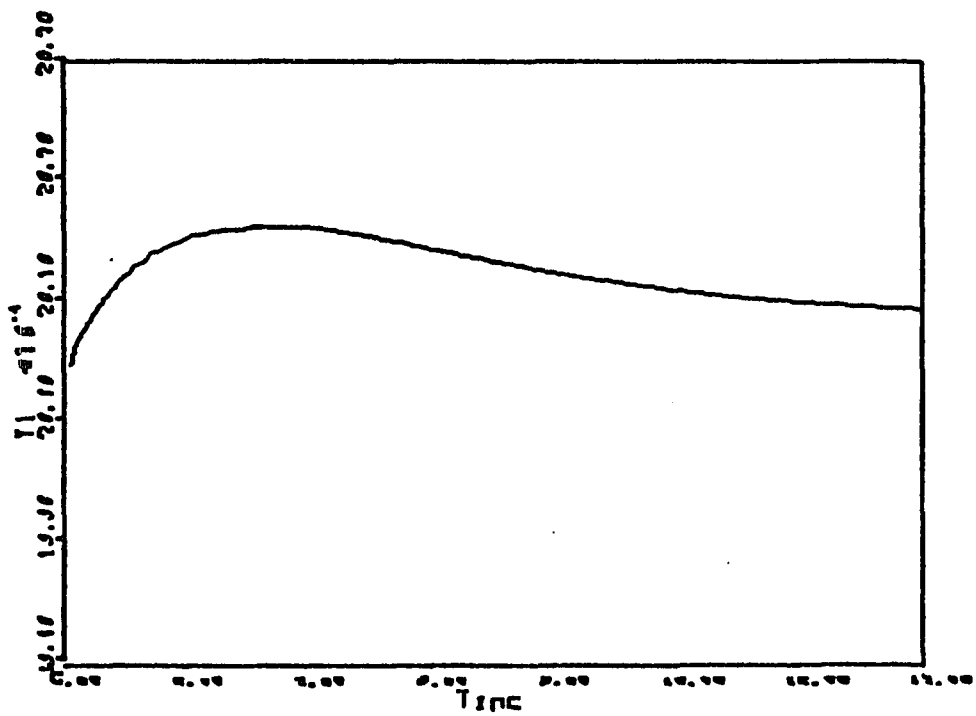
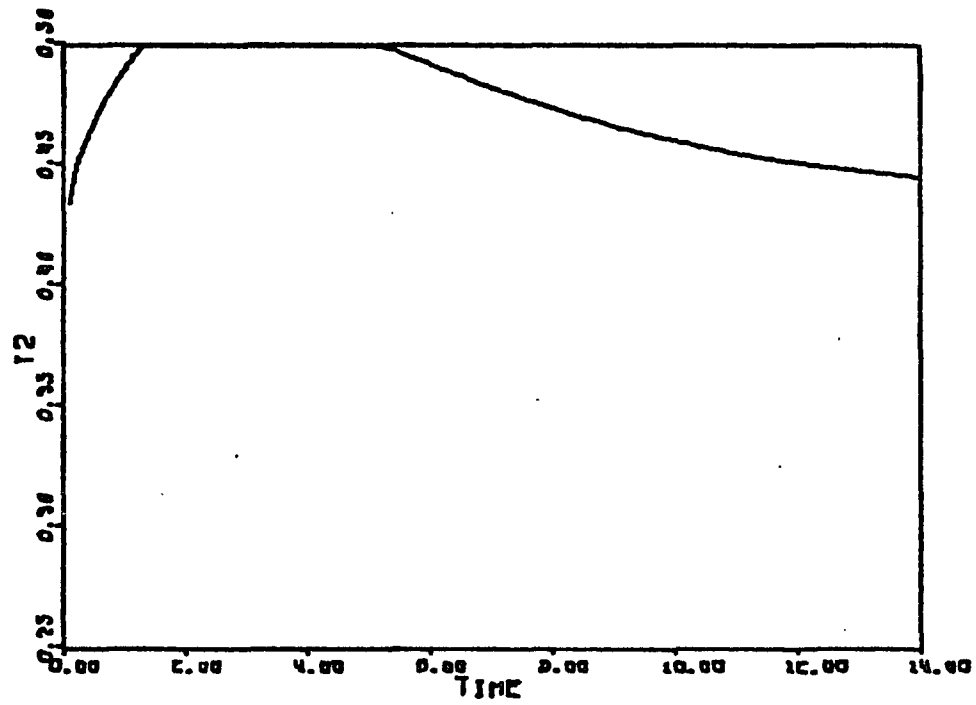


Figure 4.10. Response to step input on M_1 with steady state decoupler.

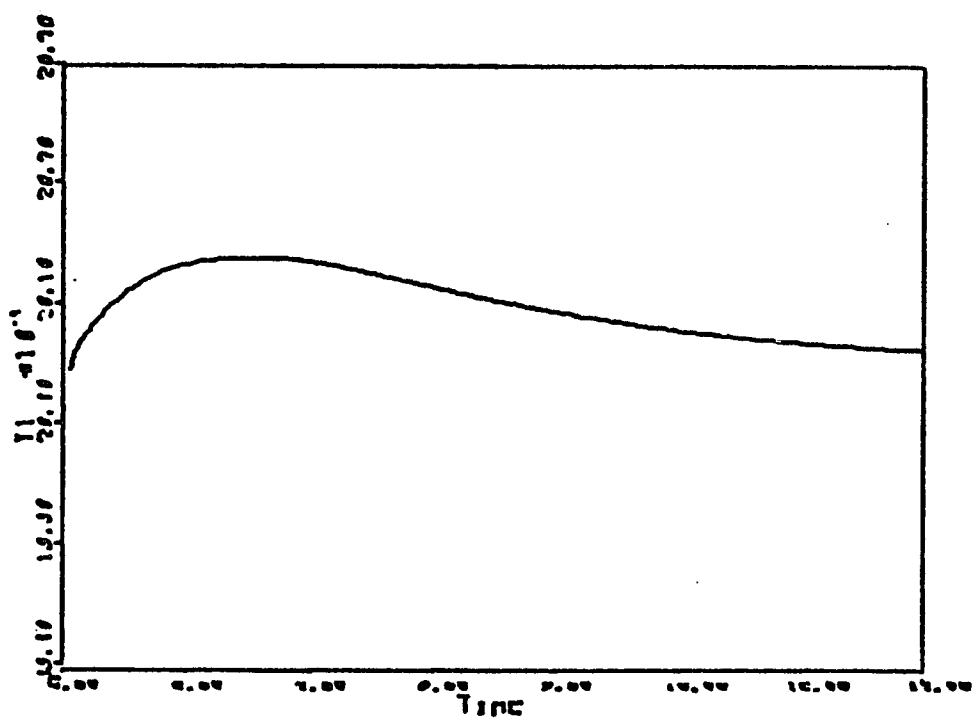
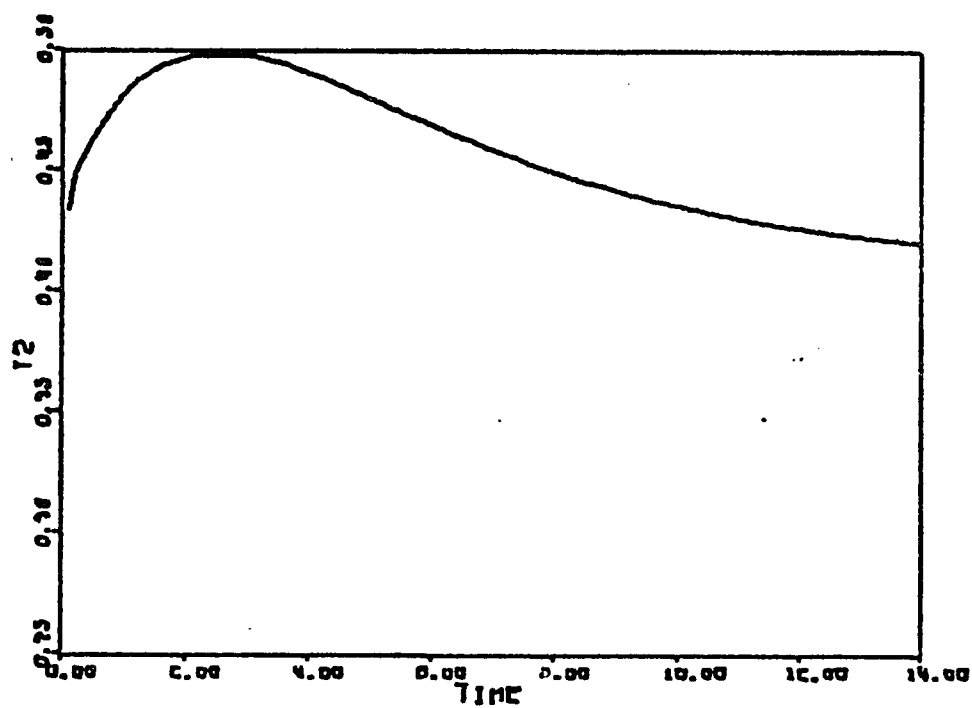


Figure 4.11. Response to step input on M_2 with steady state decoupler.

controlled variable moving in the opposite direction that the change had been made. This type of reaction tells the controller designer that this loop cannot be controlled with the typical configuration no matter what values for the controller settings are chosen. It is apparent that steady state decoupling produces as unacceptable results as no decoupling at all for this particular case. For this reason, a dynamic decoupler was designed in an attempt to compensate for the time effects of the system. Since the process is modeled as two independent loops coupled by two interacting terms, dynamic compensators only have to be placed on the terms in the decoupler that correspond to the interacting terms of the process. Each diagonal term in the decoupler, Figure 4.12, should contain a lead-lag compensator with the lead equal to the lag of its corresponding process interaction term and lag equal to the lag of the open loop for that particular manipulated variable. In addition, the diagonal term in the decoupler corresponding to the faster process interaction must be delayed by the difference between the dead times of the process interaction. In other words, the dynamics of each interaction term in the decoupler should compensate for the dynamics of the interaction in the process. The responses, Figures 4.13 and 4.14, of the system with dynamic decoupling due to step inputs for M_1 and M_2 proved to be far superior to those resulting from steady state decoupling. When one of the inputs was perturbed, its corresponding controlled

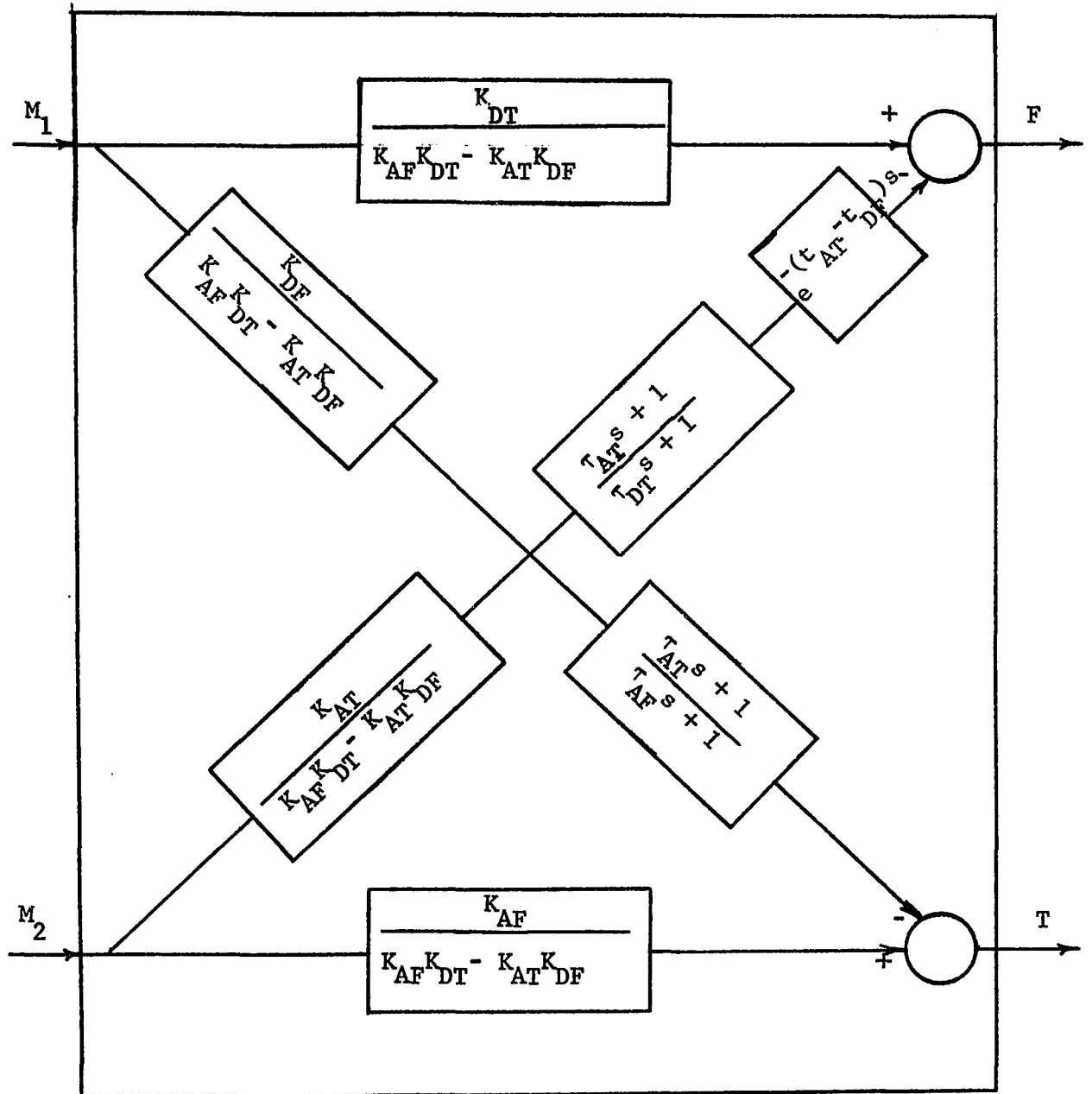


Figure 4.12. Dynamic decoupler.

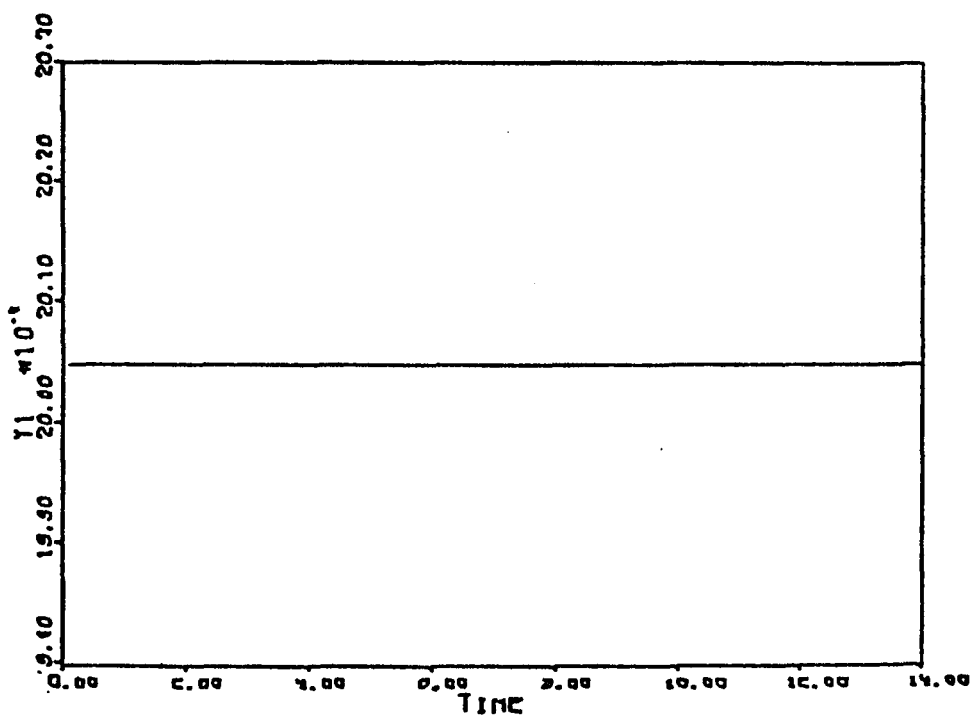
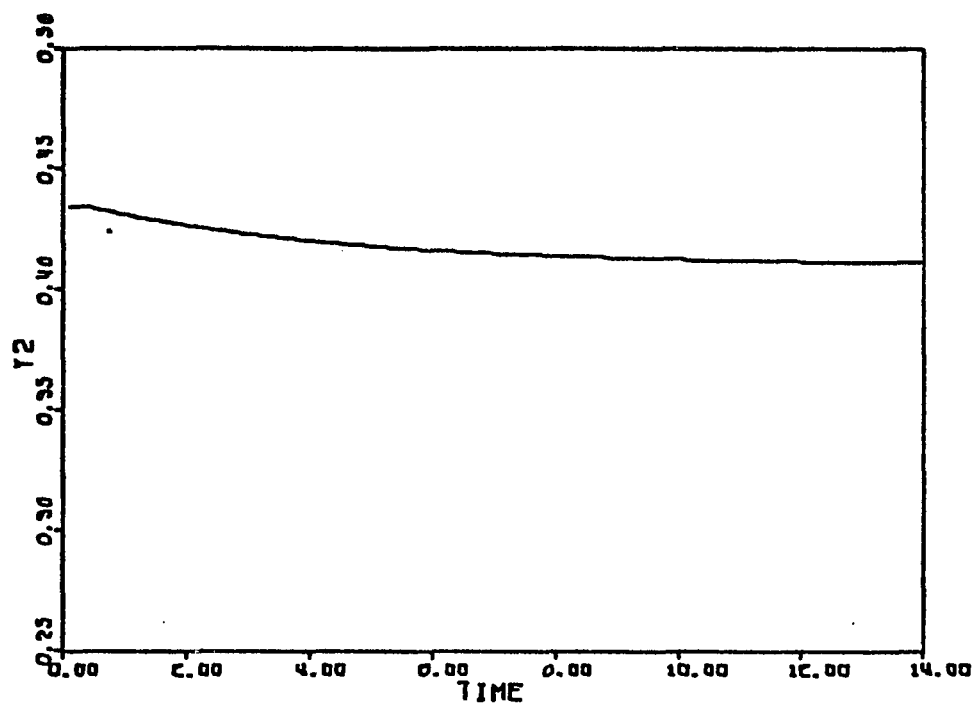


Figure 4.13. Response to step input on M_1 with dynamic decoupler.

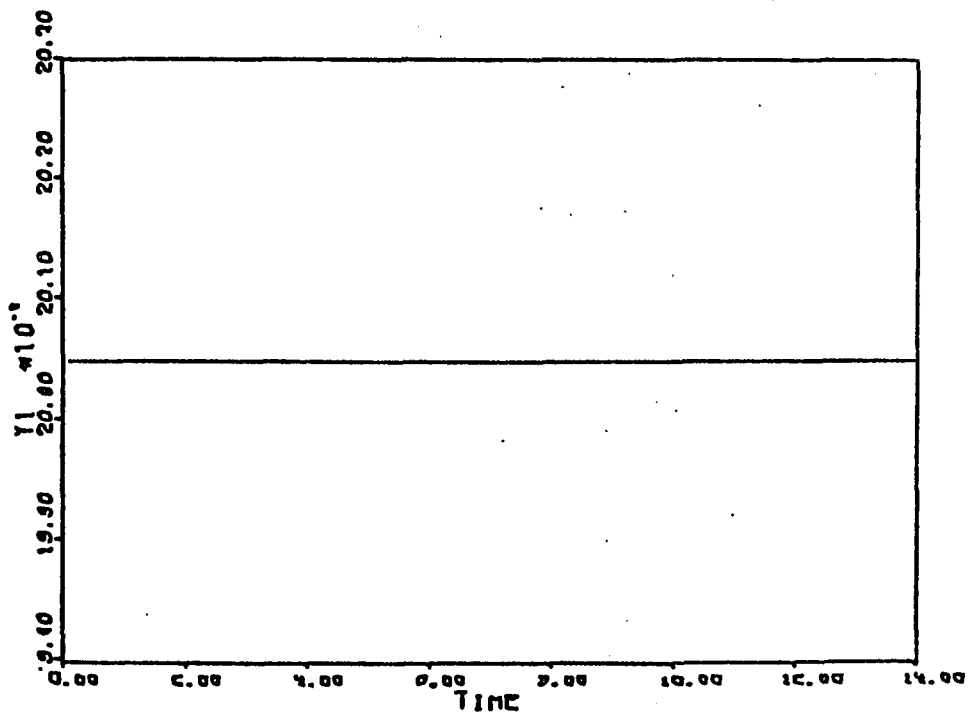
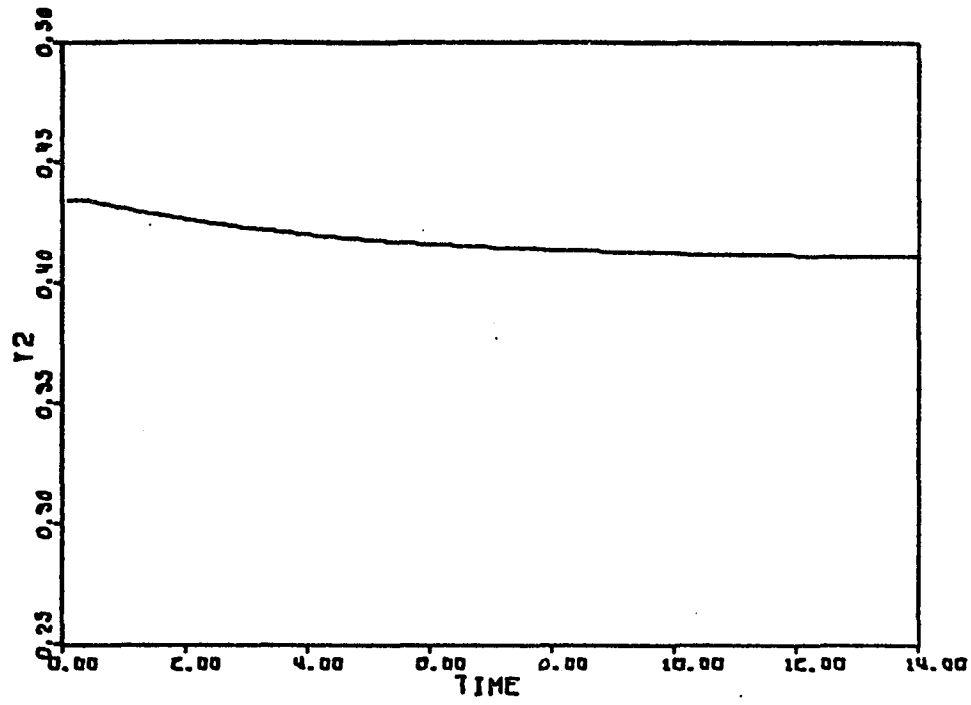


Figure 4.14. Response to step input on M_2 with dynamic decoupler.

variable approached rapidly its new value while the other controlled variable remained approximately at its previous steady state value. This is the type of response that was desired in order that the two loops could be controlled efficiently. This was substantiated when each of the loops were closed separately. Figures 4.15 and 4.16 display the fact that for a set point change the closed loop variable reaches the desired value quickly while the open loop variable is hardly affected. As can be seen in the figures, Y2 does deviate somewhat which is caused by its loop reacting slower due to larger time constants and dead times. The controller settings for each of the single loop systems were found by trial and error until they produced satisfactory results. Those settings were rather loose with the response containing no overshoot. Using those same controller constants, both loops were closed. The responses Figure 4.17-4.19 were stable and resembled those obtained when the two loops were closed separately. This suggested that if effective dynamic decoupling can be obtained, the controller settings for each loop may be found by tuning each loop separately.

Finally, Shinskey's proposal to tune controllers by compensating for interacting loops was attempted on this system. He recommends that each controller gain for each closed loop be multiplied by its corresponding term in the relative gain matrix. However, for this case since all the relative gains (absolute values) are greater than unity, an increase in gains

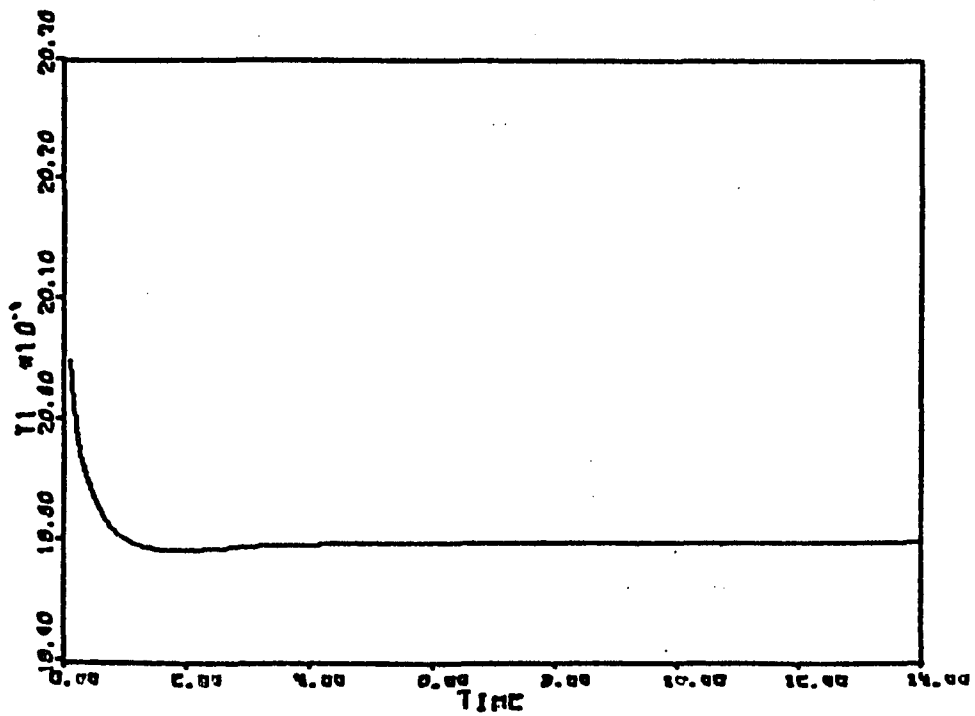
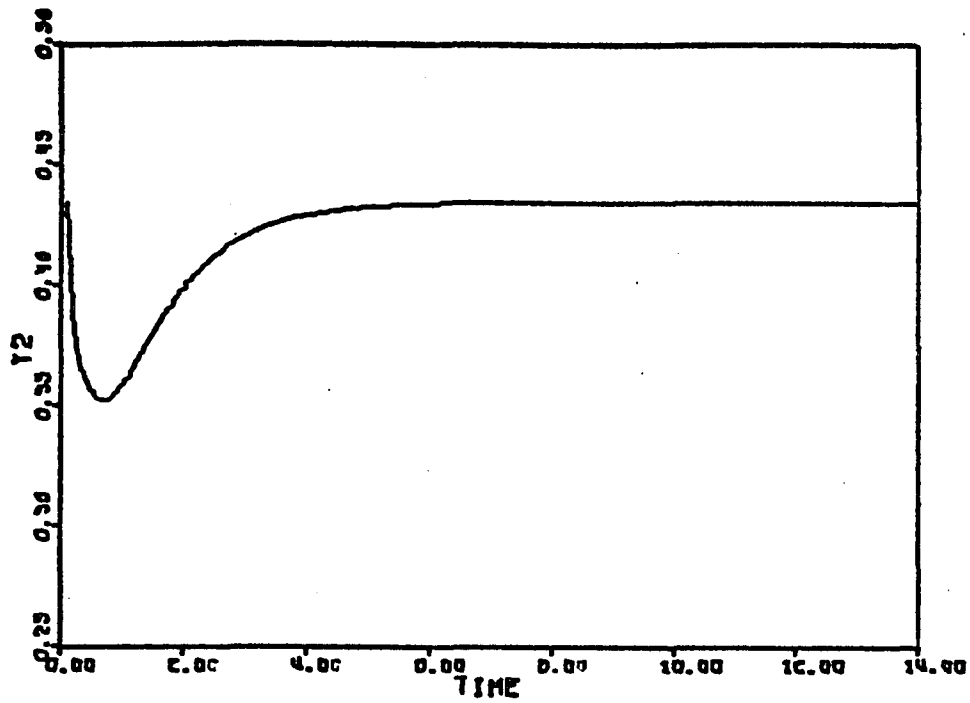


Figure 4.15. Set point response for closed loop on Y1-F with dynamic decoupler.

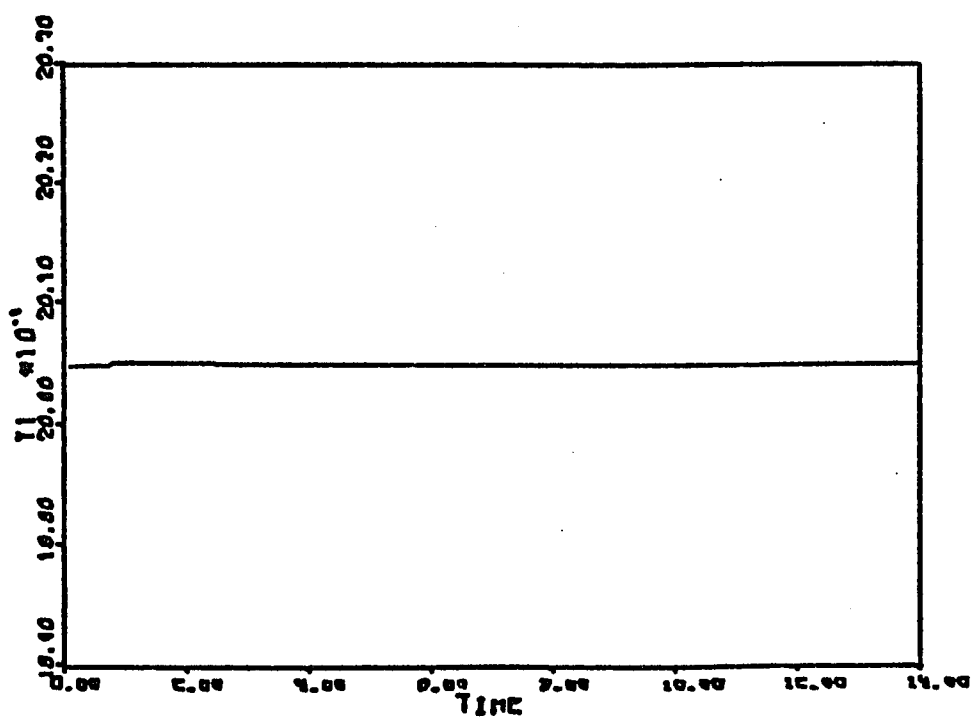
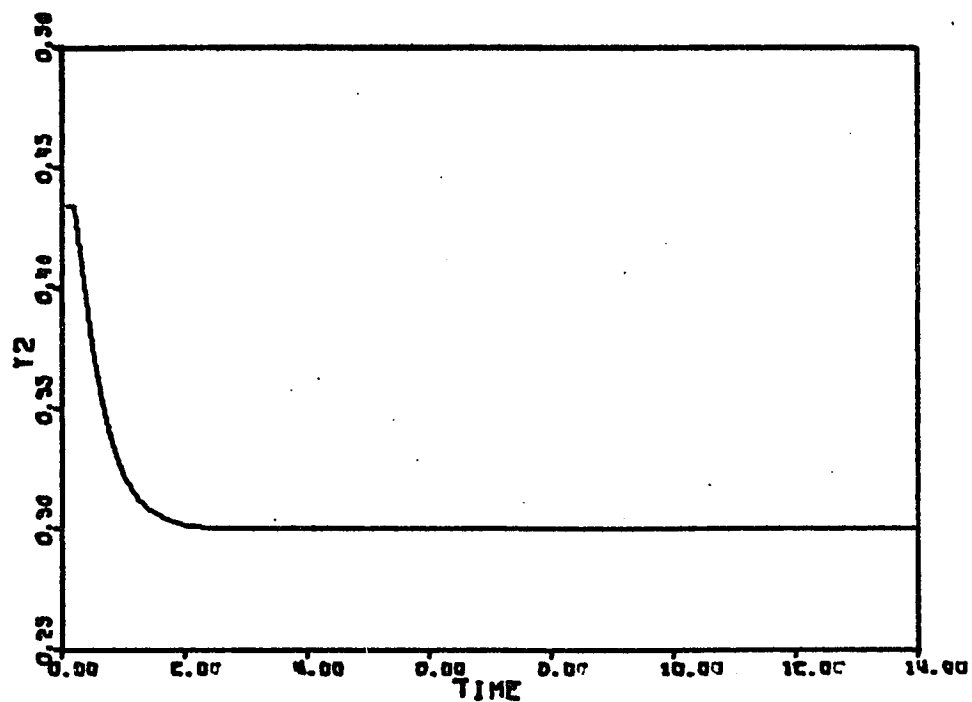


Figure 4.16. Set point response for closed loop on Y2-T with dynamic decoupler.

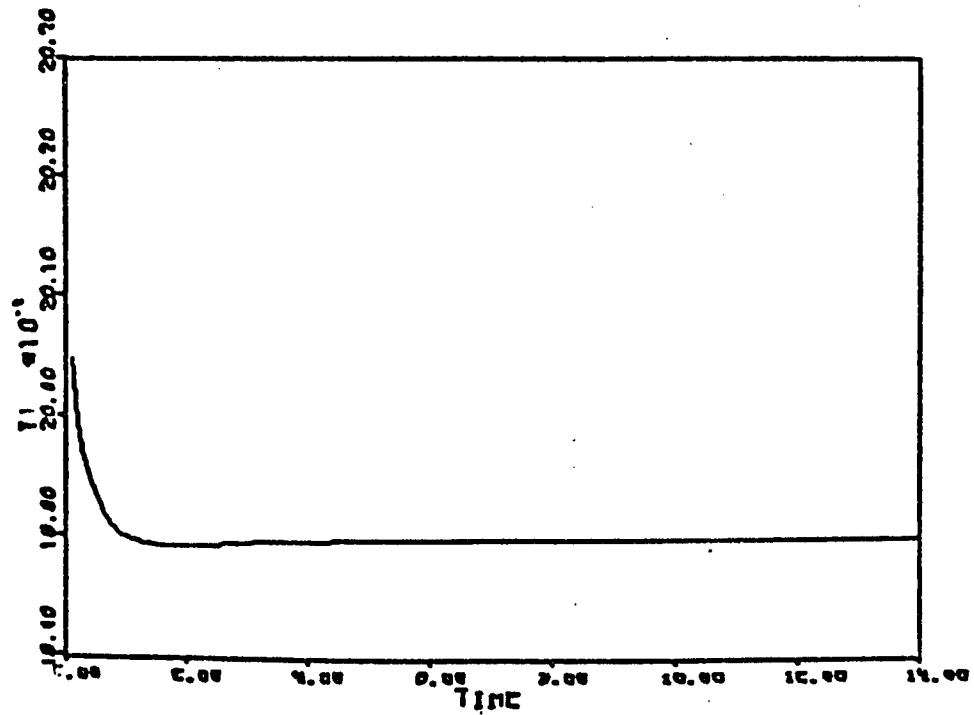
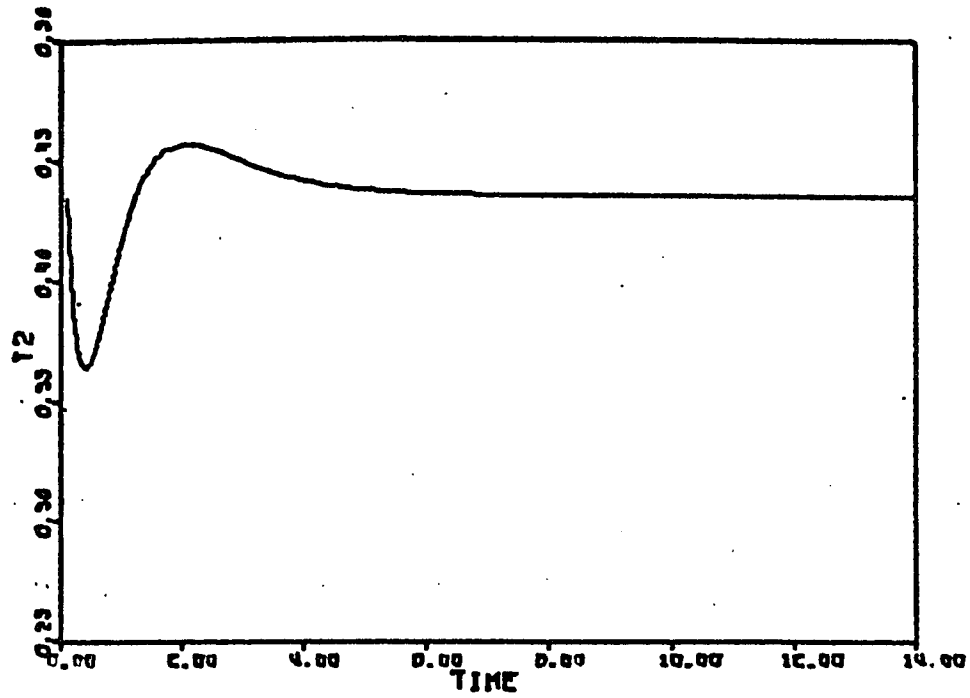


Figure 4.17. Response to set point change on Y1 with both loops closed and dynamic decoupler.

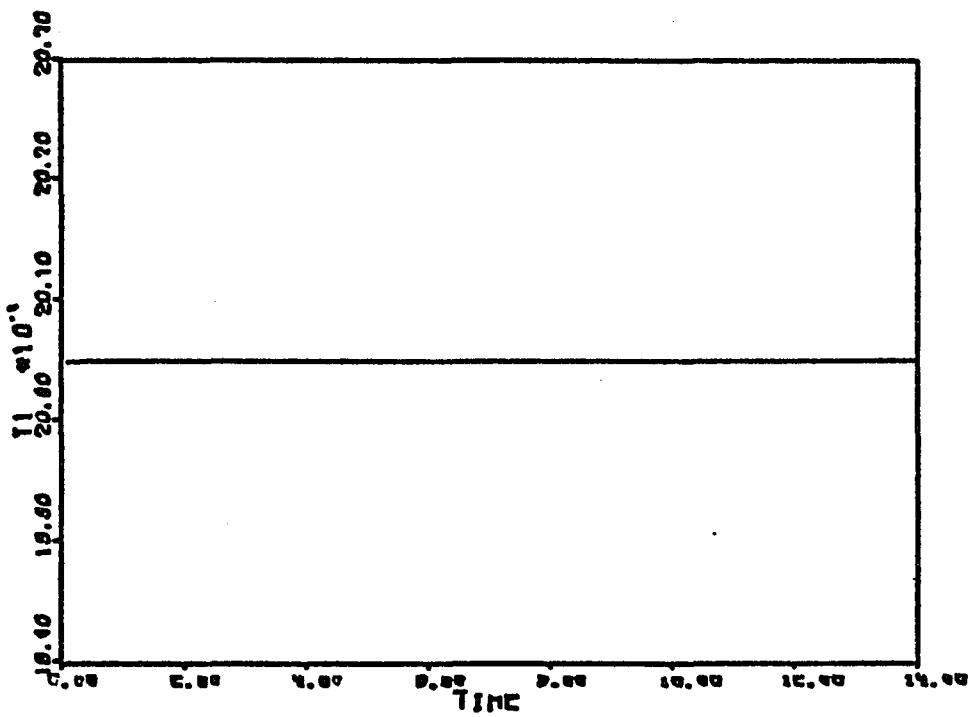
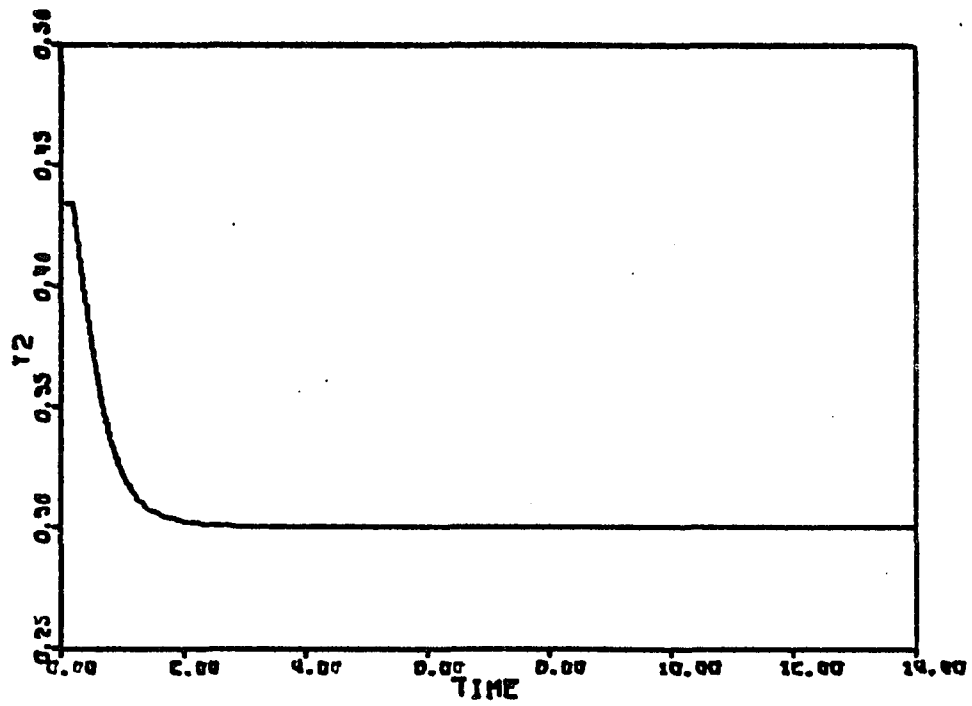


Figure 4.18. Response to set point change on Y2 with both loops closed and dynamic decoupler.

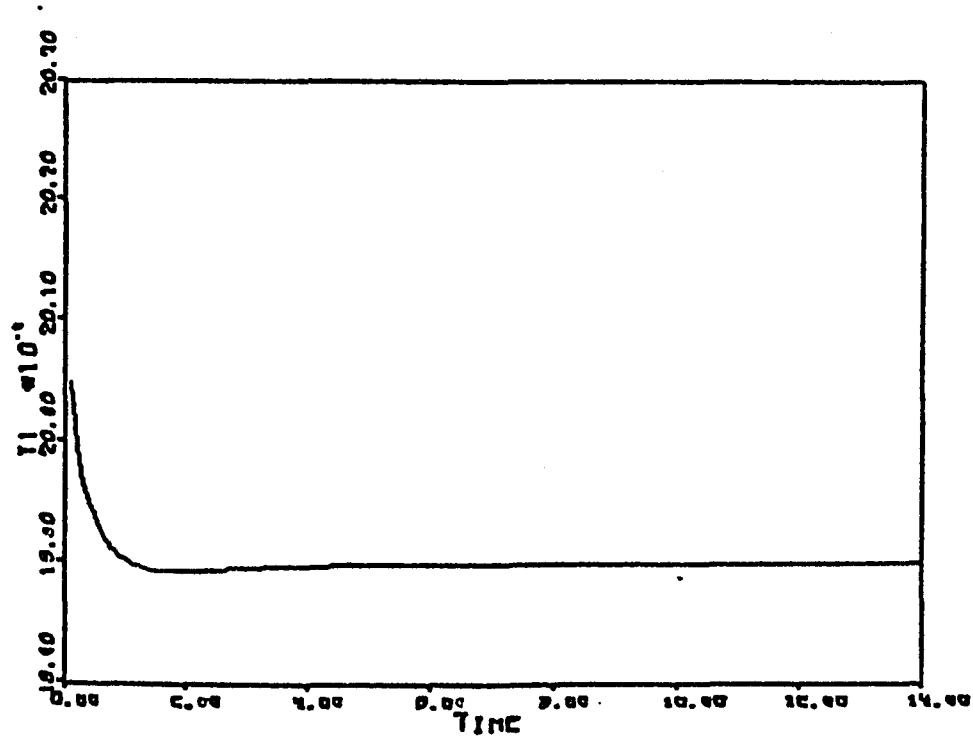
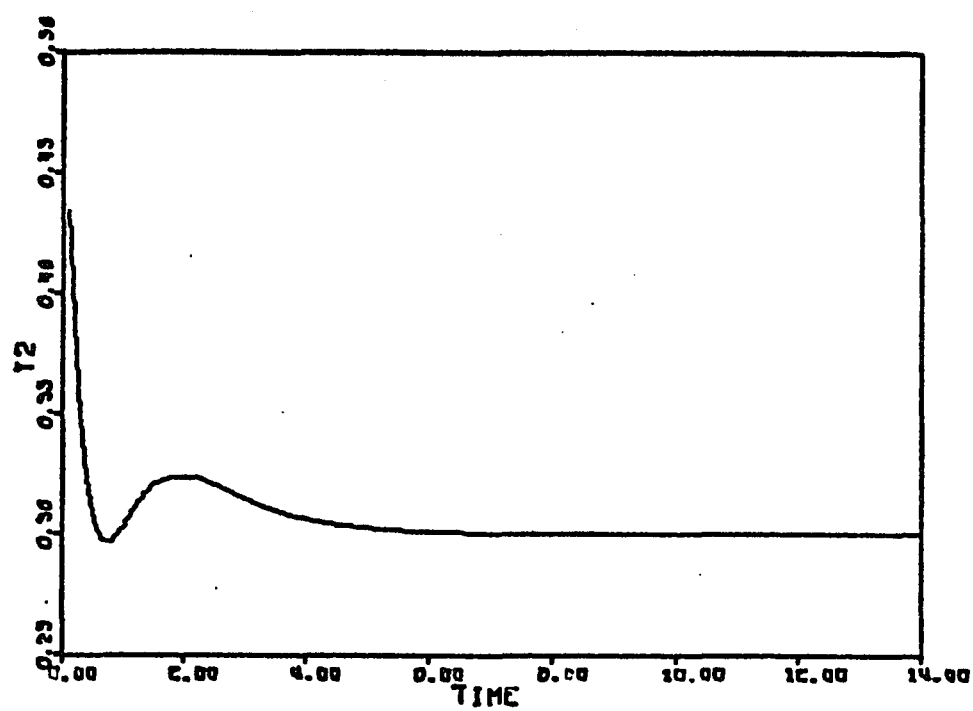


Figure 4.19. Response to change in set point on Y1 and Y2 with dynamic decoupler.

would result. Unstable responses resulted which substantiates the fact that the technique holds only if all relative gains are between zero and one.

Summary

The relative gain matrix and the CHESS simulation program are two tools for designing control schemes for complex chemical processes. The relative gain matrix along with the interaction index are two measures that can be applied to a multivariable system to properly pair manipulated and controlled variables. Both can be directly calculated from either open or closed loop steady state gains by simple experimental or analytical techniques. This chapter uses CHESS to demonstrate how the interaction measures can be employed to a chemical process. Although the CHESS system is quite large, it contains a complete thermodynamic package, many of the unit operations, a convenient data handling approach, provisions for additions to suit users particular needs, and is relatively easy to use. The approach of incorporating these two tools might prove helpful in more complex processes in which there are far more variables to consider and there exist numerous possible control schemes.

Since both the relative gain matrix and CHESS were designed for steady state conditions, their applications to dynamic situations is restricted. Also, for interacting multivariable control,

the addition of a steady state decoupler to the system will hinder the effect of the controllers causing the system to be more unstable. On the other hand, the addition of a dynamic decoupler will produce results far superior to systems with no decoupler.

Nomenclature

A	System or process matrix
C_i	Controlled or dependent variables
Det	Determinant of open loop gain matrix
$G_{ij}(s)$	Process transfer function
I_{ij}	Interaction index
K	System open loop gains
K_c	Controller gains
\underline{M}	Open loop gain matrix
M_j	Manipulated or independent variables
T_i	Controller reset time
t_o	Process dead time
ϵ	Disturbance or upset
λ_{ij}	Elements of the relative gain matrix
τ	Process time constant

Literature Cited

1. Bristol, E. H. , "On a Measure of Interaction for Multi-variable Process Control", IEEE Transactions on Automatic Control, Jan. 1966, pp. 133-134.
2. Corripio, Armando B., Notes on Multivariable Process Control, Louisiana State University, Baton Rouge, La., 1971.
3. Juantorena, R., and Romeo, R. J., "Application of the Relative Gain Matrix to a Distillation Column", Proceedings of the ISA AID/CHEMPID Symposium, Houston, Texas, April 1971.
4. Miller, John A., Ph.D. Dissertation, Louisiana State University, Baton Rouge, La. 1970.
5. Motard, R. L., System Guide for CHESS, Department of Chemical Engineering, University of Houston, 1968.
6. Murrill, P. W., Automatic Control of Processes, International Textbook Co., Scranton, Penn. 1967.
7. Nisenfeld, A. E., and Schultz, H. M., "Interaction Analysis in Control System Design", Paper 70-562, Proceedings of the 25th ISA Conference, Advance in Instrumentation, Vol. 25, 1970.
8. Nisenfeld, A. E., and Stravinski, C. A., "Feedforward Control in Azeotropic Distillations", Chemical Engineering, Sept. 23, 1968, pp. 227-236.
9. Shinskey, F. G., Process Control Systems, McGraw-Hill Book, Company, New York, 1967, pp. 181-203.
10. Shinskey, F. G., "Stable Distillation Control Through Proper Pairing of Variables", Proceedings of the ISA AID/CHEMPID Symposium, Houston, Texas, April 1971, pp. 36-40.

Chapter V

CONCLUSIONS

This work is intended to present some of the concepts of advanced control theory and the modifications that they must undergo before they can be applied in the process industries. Because these industrial processes cannot be easily described mathematically, much of the theory cannot be used directly and approximations have to be employed. This thesis is written to give the control system designer for a chemical process some insight into the problems of applying the theory developed for other systems to his process and some of the modifications or approximations that might prove acceptable.

In Chapter II emphasis was on model identification. Sensitivity coefficients were used to determine and update model parameters in order to dictate the control strategy for a chemical reactor process. The adaptive control scheme worked well and this approach seemed suited for on-line applications. With the sensitivity coefficients adapting a second order model, three control strategies were implemented. A conventional PI analog controller which was optimally tuned produced the expected satisfactory responses. The Kalman algorithm was then inserted

with its controller constants calculated from the model parameters. The method was modified somewhat due to the limitations placed on the manipulated variable. The responses of the process, with the Kalman algorithm, were improved appreciably over those of the PI controller. Finally, a bang-bang controller which resulted from a time optimal approach was illustrated. The switching times were determined from the model parameters and the responses of the chemical reactor due to set point changes were superior to those responses of the first two techniques. Thus, when only small parameter changes are expected, the coefficients will accurately predict model parameters. However, if the parameters differ greatly from their true values, instabilities may result. Also, in certain regions the coefficients do not determine a true solution, but oscillate between parameter values as in some gradient techniques. These problems may be overcome by means of special methods such as the steepest descent approach as proposed by Marquardt.

Chapter III showed that optimal control can be applied directly to processes modeled with a first order lag plus dead time in order to obtain controller settings for a PID analog controller. Although the responses were suboptimal due to modeling errors, the results for this method for a chemical reactor process appeared better than those tuned by other techniques. This approach allows the user to choose a performance

criterion to suit his needs; that is, the value of the parameter p in the cost function will either tighten up or loosen the control. In other words, the tuning problem is reduced from a three parameter to a one parameter search.

The PCID algorithm was tested and found to produce more consistent results for both the load and set point problem than the conventional PID algorithm. Controller settings for the PCID resulted in satisfactory responses for both load and set point changes while the PID had to be tuned separately for each case.

The extension of the optimal control formulation to the sampled-data system was found to be unacceptable. By approximating a sampled-data system with a continuous one by adding a pure time delay equal to one-half the sampling time to the system dead time, it was thought that the responses for the two systems should be similar. This was not the case however, and only at extremely fast sampling rates did the scheme achieve satisfactory responses.

Chapter IV discusses some of the aspects of multivariable control. The first step in the design of any control system is to determine whether it is possible for the process to be controlled. The proper pairing of controlled and manipulated variables is essential for controllability in some cases and in the others there is probably a preferred configuration. The relative process gain matrix will determine the correct

pairing of variables for all cases and whether both dynamic and steady state control are possible. For an absorber-stripper process, CHESS was used to calculate the steady state open loop gains. From this matrix the relative gains were found and the effects of different manipulated and controlled variables could be observed.

Dynamic considerations for interacting loops were studied by means of a general model with two inputs and two outputs. A steady state decoupler incorporated into the system produced responses that were worse than those resulting from the system with no decoupler. This illustrates the fact that the time effects of the process had to be considered in order to achieve effective decoupling. By adding a dynamic decoupler, the interacting terms were almost canceled and two nearly independent loops were obtained. Independent loops can be tuned individually whereas the loops for the interacting system are almost impossible to tune.

The advanced tuning relationships proposed by Shinskey for interacting loops were applied and found to be unsuccessful in the case investigated. When elements of the relative gain matrix are greater than one (less than zero) the concept of compensating controller settings for variable interaction does not hold.

APPENDIX A
DEVELOPMENT OF CONTROL LAW USING
KALMAN'S ALGORITHM

APPENDIX A
DEVELOPMENT OF CONTROL LAW USING
KALMAN'S ALGORITHM

Kalman proposes that a controller be designed that will drive the system to the desired value in two sampling periods. This is accomplished by imposing a large acceleration, forcing function on the first sampling period and then applying a decelerating force so that there will be no overshoot in the response of the process. Optimal control theory states that for a second order system, time-optimal control will contain two switching times. Thus, for this case a reasonable response curve could be expected. A diagram of control strategy and response for a normalized system due to a set point change is given below. Notice that at $t=0$, the controller sets the manipulated value to M_1 . If no further changes were applied, the response would appear as shown in the dotted line. At T_1 the controller output switches to M_2 causing the system response to slow down as shown by the solid line. Finally, the manipulated variable reaches the steady state value at $t=T_2$ at which time the system response is leveling at the desired value. This will be the type of response that the Kalman controller will ordinarily produce. The mathematic description of the process is given by the general second order model equation

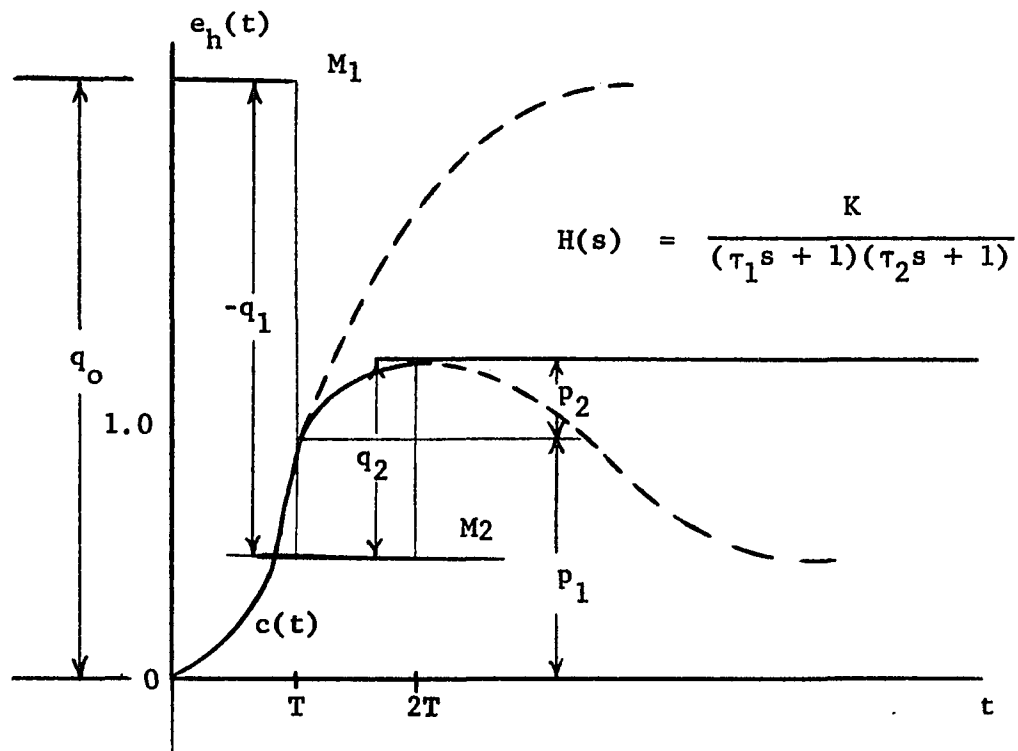


Figure A2.1. Sketch of Ideal Response for Kalman Algorithm

$$H(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (\text{A1.1})$$

Since this will be a discrete or sampled-data control system a zero order hold will be incorporated with the process equation. The model for the entire plant in terms of z-transforms then becomes

$$G^*(z) = (1 - z^{-1}) H^*(s) / s \quad (\text{A1.2})$$

By writing equations for closed loop inputs and outputs the following relationships can be obtained (refer to Figure 2.5)

$$\frac{M^*(z)}{C^*(z)} = \left[\frac{D^*(z)}{1 + D^*(z)G^*(z)} \right] = Q^*(z) \quad (\text{A1.3})$$

and

$$\frac{C^*(z)}{R^*(z)} = \frac{D^*(z)G^*(z)}{1 + D^*(z)G^*(z)} = P^*(z) \quad (\text{A1.4})$$

These two equations lead to the relationship that

$$\frac{C^*(z)}{M^*(z)} = G^*(z) = \frac{P^*(z)}{Q^*(z)} \quad (\text{A1.5})$$

Steady state considerations force $\sum p_i$ and $\sum q_i$ to equal 1. By choosing $H(0)=1$ and restricting $G^*(z) = 1$ then $D^*(z)$ may be written as

$$D^*(z) = \frac{Q^*(z)}{1 - G^*(z)Q^*(z)} = \frac{Q^*(z)}{1 - P^*(z)} \quad (\text{A1.6})$$

The above equation yields the controller coefficients for the Kalman algorithm. The procedure is simply to find $G^*(z)$ and multiply numerator and denominator by a suitable scale factor forcing the sum of the p 's to equal one. Then equation A1.6 will determine the controller coefficients. The following example is for a general second order system with transfer function $H(s)$.

$$H(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (A1.7)$$

$$G^*(z) = \left[\frac{z-1}{z} \frac{H(s)}{s} \right]^* \quad (A1.8)$$

$$G^*(z) = \left[\frac{z-1}{z} \frac{K/\tau_1 \tau_2}{s(s+1/\tau_1)(s+1/\tau_2)} \right]^* \quad (A1.9)$$

From a table of z -transforms if $E(s) = 1/s(s+a)(s+b)$ then

$$E(z) = \frac{1}{ab} \left[\frac{1}{z-1} + \frac{bz}{(a-b)(z-e^{-aT})} - \frac{az}{(a-b)(z-e^{-bT})} \right] \quad (A1.10)$$

then if $a = 1/\tau_1$ and $b = 1/\tau_2$

$$G^*(z) = \frac{\tau_1 \tau_2 K}{\tau_1 \tau_2} \left[\frac{z-1}{z} \right] \left[\frac{z}{z-1} + \frac{\tau_1 \tau_2 z}{\tau_2 (\tau_2 - \tau_1) (z-e^{-T/\tau_1})} - \frac{\tau_1 \tau_2 z}{\tau_1 (\tau_2 - \tau_1) (z-e^{-T/\tau_2})} \right]$$

$$= K(z-1) \left[\frac{1}{z-1} + \frac{\tau_1}{(\tau_2 - \tau_1)(z - e^{-T/\tau_1})} - \frac{\tau_2}{(\tau_2 - \tau_1)(z - e^{-T/\tau_2})} \right] \quad (\text{A1.11})$$

If these terms are combined and the coefficients of the z terms lumped together $G^*(z)$ will become

$$G^*(z) = \frac{K}{\tau_2 - \tau_1} \left[\frac{(\tau_2 - \tau_1)(z - e^{-T/\tau_1})(z - e^{-T/\tau_2}) + \tau_1(z-1)(z - e^{-T/\tau_2})}{(z - e^{-T/\tau_1})(z - e^{-T/\tau_2})} - \frac{\tau_2(z-1)(z - e^{-T/\tau_1})}{(z - e^{-T/\tau_1})(z - e^{-T/\tau_2})} \right] \quad (\text{A1.12})$$

$$G^*(z) = \frac{A_1 z + A_2}{z^2 + B_1 z + B_2} = \frac{A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}} \quad (\text{A1.13})$$

where

$$\begin{aligned} A_1 &= \frac{K}{\tau_2 - \tau_1} \left[(\tau_2 - \tau_1)(-e^{-T/\tau_1} - e^{-T/\tau_2}) - \tau_1(1 + e^{-T/\tau_2}) + \tau_2(1 + e^{-T/\tau_1}) \right] \\ A_2 &= \frac{K}{\tau_2 - \tau_1} \left[(\tau_2 - \tau_1)(e^{-T/\tau_1 - T/\tau_2}) + \tau_1 e^{-T/\tau_2} - \tau_2 e^{-T/\tau_1} \right] \\ B_1 &= -e^{-T/\tau_1} - e^{-T/\tau_2} \\ B_2 &= e^{-T/\tau_1 - T/\tau_2} \end{aligned}$$

Forcing the sum of the numerator of $G^*(z)$ to one

$$A_1 N + A_2 N = 1 \text{ implies } N = 1/(A_1 + A_2)$$

Multiplying numerator and denominator of equation A1.13 by N:

$$G^*(z) = \frac{\alpha_1 z^{-1} + \alpha_2 z^{-2}}{\beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2}} \quad (\text{A1.14})$$

and

$$D^*(z) = \frac{M(z)}{E(z)} = \frac{\beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2}} \quad (\text{A1.15})$$

where

$$\beta_0 = N$$

$$\beta_1 = B_1 N$$

$$\beta_2 = B_2 N$$

$$\alpha_1 = A_1 N$$

$$\alpha_2 = A_2 N$$

The control law can then be written

$$M_N = \beta_0 E_N + \beta_1 E_{N-1} + \beta_2 E_{N-2} + \alpha_1 M_{N-1} + \alpha_2 M_{N-2} \quad (\text{A1.16})$$

This equation shows that the controller uses the present and last two values of the error signal along with the last two values of the manipulated variable in order to calculate the new value for the control variable. For the case where the manipulated variable has limits, if the new value is outside the allowed range the calculated value is replaced by the limit. However,

for the algorithm to work at the next sampling time, a pseudo-error signal determined by solving equation A1.16 for E_N with M_N replaced by the limit. This calculated value of E_N is stored and used for further calculations. The procedure works well for the case investigated.

APPENDIX B

SUM OF ROWS AND COLUMNS OF
RELATIVE GAIN MATRIX EQUAL ONE

APPENDIX B

SUM OF ROWS AND COLUMNS OF
RELATIVE GAIN MATRIX EQUAL ONE

An important feature of the relative gain matrix is that the sum of any row or column is equal to one. Matrix algebra will show that this is true for any system. For example, the system equations for a process with three inputs and three outputs may be written as

$$\underline{c} = \underline{A} \underline{M} \quad (\text{A2.1})$$

where

\underline{c} = system output, controlled variables

\underline{M} = system input, manipulated variables

\underline{A} = system matrix representing system equations

These equations may be linearized around an operating point and the process equations be written as

$$dC_1 = A_{11}dM_1 + A_{12}dM_2 + A_{13}dM_3$$

$$dC_2 = A_{21}dM_1 + A_{22}dM_2 + A_{23}dM_3$$

$$dC_3 = A_{31}dM_1 + A_{32}dM_2 + A_{33}dM_3 \quad (\text{A2.2})$$

or

$$\underline{dC} = \underline{A} \underline{dM} \quad (\text{A2.3})$$

where

dC = changes from operating point of output

dM = changes from operating point of input

A = a constant 3×3 matrix

By applying the definition of the open loop or steady state gains, the M matrix can readily be written as

$$M = \begin{bmatrix} \left. \frac{dC_1}{dM_1} \right|_M & \left. \frac{dC_1}{dM_2} \right|_M & \left. \frac{dC_1}{dM_3} \right|_M \\ \left. \frac{dC_2}{dM_1} \right|_M & \left. \frac{dC_2}{dM_2} \right|_M & \left. \frac{dC_2}{dM_3} \right|_M \\ \left. \frac{dC_3}{dM_1} \right|_M & \left. \frac{dC_3}{dM_2} \right|_M & \left. \frac{dC_3}{dM_3} \right|_M \end{bmatrix} \quad (\text{A2.4})$$

From an examination of the system equations and also the definition for the steady state gains, it can be seen that the partial derivatives may be replaced by the elements of the A matrix. The first system equation can be used to solve for the first element of the M matrix which is defined as $\left. \frac{dC_1}{dM_1} \right|_M$. Since

$$dC_1 = A_{11} dM_1 + A_{12} dM_2 + A_{13} dM_3 \quad (A2.5)$$

and M_2 and M_3 being constant imply dM_2 and dM_3 equal zero so that $dC_1 = A_{11} dM_1$ or

$$\left. \frac{dC_1}{dM_1} \right|_{M_2 M_3} = A_{11} \quad (A2.6)$$

Likewise, the remainder of the M matrix is given by

$$M = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad (A2.7)$$

Since this matrix is a constant, the relative gain matrix will also be constant and is calculated directly from the above equations which are the open loop gains. The closed loop gains can be calculated from the system equations by means of Cramer's Rule. Rearranging,

$$\underline{M} \underline{\Delta m} = \underline{\Delta c} \quad (A2.8)$$

The gain of each controlled variable can be found by holding the other two outputs constant, setting Δc_2 and Δc_3 equal to zero.

Applying Cramer's Rule to the three system equations and solving for Δm_1 , Δm_2 , and Δm_3 , yield the closed loop gains for c_1 . For example, $\Delta c_1 / \Delta m_1 \big|_{c_2, c_3}$ is found by the following series of operations:

$$\Delta m_1 = \frac{\det \begin{bmatrix} \Delta c_1 & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}}{\det \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}}$$

$$\Delta m_1 = \frac{\Delta c_1 (A_{22}A_{33} - A_{23}A_{32})}{\det(M)}$$

or

$$\left. \frac{\Delta c_1}{\Delta m_1} \right|_{c_2, c_3} = \frac{\det(M)}{A_{22}A_{33} - A_{23}A_{32}} \quad (A2.9)$$

where the determinant of M is written as

$$\begin{aligned} \det(M) = & A_{11}A_{22}A_{33} - A_{11}A_{23}A_{32} - A_{21}A_{12}A_{33} + A_{21}A_{13}A_{32} \\ & + A_{31}A_{12}A_{23} - A_{31}A_{13}A_{22} \end{aligned} \quad (A2.10)$$

Similarly the remainder of the closed loop gains can be obtained and the entire array is given as

	Δm_1	Δm_2	Δm_3
Δc_1	$\frac{\det(M)}{A_{22}A_{33} - A_{23}A_{32}}$	$\frac{\det(M)}{A_{23}A_{31} - A_{21}A_{33}}$	$\frac{\det(M)}{A_{21}A_{32} - A_{22}A_{31}}$
Δc_2	$\frac{\det(M)}{A_{13}A_{32} - A_{12}A_{33}}$	$\frac{\det(M)}{A_{11}A_{33} - A_{13}A_{31}}$	$\frac{\det(M)}{A_{12}A_{31} - A_{11}A_{32}}$
Δc_3	$\frac{\det(M)}{A_{12}A_{23} - A_{13}A_{22}}$	$\frac{\det(M)}{A_{13}A_{21} - A_{23}A_{11}}$	$\frac{\det(M)}{A_{11}A_{22} - A_{12}A_{21}}$

The relative gain matrix is determined by dividing the M matrix of equation A2.7 term by term by the matrix given above.

$$\lambda = \begin{array}{c} \left| \begin{array}{ccc} \frac{A_{11}(A_{22}A_{33} - A_{23}A_{32})}{\det(M)} & \frac{A_{12}(A_{23}A_{31} - A_{21}A_{33})}{\det(M)} & \frac{A_{13}(A_{21}A_{32} - A_{22}A_{31})}{\det(M)} \\ \frac{A_{21}(A_{13}A_{32} - A_{12}A_{33})}{\det(M)} & \frac{A_{22}(A_{11}A_{33} - A_{13}A_{31})}{\det(M)} & \frac{A_{23}(A_{12}A_{31} - A_{11}A_{32})}{\det(M)} \\ \frac{A_{31}(A_{12}A_{23} - A_{13}A_{22})}{\det(M)} & \frac{A_{32}(A_{13}A_{21} - A_{23}A_{11})}{\det(M)} & \frac{A_{33}(A_{11}A_{22} - A_{12}A_{21})}{\det(M)} \end{array} \right. \end{array}$$

Inspection of this array along with the definition of the determinant of M leads to the proof that the sum of any row or any column equals one.

APPENDIX C

USING CHESS ON THE SIGMA 5

Introduction

This section is written to be a user's guide for those desiring to run CHESS on the XDS-Sigma 5 computer with only 20 K core storage available. Because CHESS was designed to run on much larger computer configurations, many changes had to be made to decrease the memory necessary for execution. Although the basic CHESS concept remains in this version, a number of the options and flexibilities of the original package are no longer available. This will be the objective of this section, to list and explain the changes implemented. For this reason it should be considered an appendix of the CHESS system guide and used only after a basic understanding of the original version has been obtained. The original version will be considered the last configuration suggested by the University of Houston, including all changes and corrections listed in the newsletters through February 1971.

System Description

Since the CHESS system is composed of a number of sub-programs or modules (executive programs and supporting functions), a list of the changes made in some of these modules will be listed.

MAIN: The calling program for the new version has been modified because executing an overlaid program is slightly different on the Sigma 5 than on the IBM 360. First, all common blocks needed

for the system have been placed in MAIN. Although this is less efficient as far as storage locations, loading is simplified. In segmented programs, labeled common blocks are allocated in the segment in which they are first defined. As the segments are read in and out of core during execution, the allocated storage is erased unless special steps are taken. In other words, when a segment is removed from core and then read back in, all labeled common that did not remain in core will be zeroed. For this reason, the placing of all common blocks in MAIN, which is in the root segment and always present in core, was thought to be the easiest approach. Other changes in MAIN include the call statements for the needed overlay segments. These statements are of the form CALL SEGLD(I) where I is the desired segment. These statements will be discussed in more detail in the section describing overlaying on the Sigma 5.

DREAD: DREAD in the original version is really a set of sub-routines designed to read and interpret the input data supplied by the user to run a particular problem. There were two forms of input: namelist data and free-form data. Since namelist data is easily understood and used and also requires a fewer number of storage locations, it was decided to delete the free-form interpreter. Although this eliminates some of the flexibility of the original system, it does not reduce any of the system capabilities. Omission of the free-form interpreter allows the

deletion of subroutines INCR, INTNUM, NUM, RBLN and RELNUM. Another change in DREAD is in the namelist read statement. The normal Fortran IV read statement is of the form READ(5,XYZ) where XYZ is a namelist title defined by NAMELIST/XYZ/I,J,X(2,3). The input data must be of the form

```
XYZ I = 1, J = 2, X(2,3) = 4.0, & END
```

When a read namelist statement is encountered, the machine will read through the data cards until the referenced name (XYZ) is read. It then reads all information until the &END statement is processed. (See Fortran IV manual for complete details). The Sigma 5 handles namelist data slightly different. Although no namelist titles are defined as in Fortran IV, all variables to be read in through a namelist call must be so designated by means of a statement of the form NAMELIST I,J,X(2,3). The read statement is simply Input(5) where 5 denotes the card reader, and the machine then reads all input information until an asterisk (*) is encountered. This difference forces the user to be aware of the form and order of the data but it is no real problem. However, these are two major restrictions of the Sigma 5 compiler that limits the effectiveness of namelist input. First, no alphanumeric information can be read with the input statement. This affects directly the entering of equipment module names and

external names in the KPM matrix. In order to obtain the correct naming of equipment, a code number is used instead of the alphanumeric name. The following code numbers are entered in the namelist data.

<u>Equipment Name</u>	<u>Code Number</u>
Divider	1
Distillation Column	2
Mixer	3
Adiabatic Flash Unit	4
Reactor	5
Valve	6
Heat Exchanger	7
Pump	8
Absorber	9
Multistage Equilibrium Unit	10
Fired Heater	11

The code number will appear in the second position of the KPM matrix in place of the actual equipment name. In addition, the external name will automatically be set equal to the equipment name and its position in the input array has been deleted, decreasing the size of the KPM matrix from 10 to 9. For example, if a pump is desired, the user would write

KPM16 = 16, 8, 40,-15,0,0,0,0,0,

instead of

KPM16 = 16, 'PUMP', 'P-2', 40,-15,5*0,

In the above example the second restriction of the Sigma 5 compiler can be seen. Because the asterisk denotes end of data set, it cannot be used to denote repeating data. In order to zero the remainder of an input array, the correct number of zeros have to be punched on the data cards. This is not as convenient as the normal system in which an asterisk and integer denotes the number of times a value is to be entered consecutively in a data set. For example, if the variable A is dimensioned as A(10) and it is desired to set the first value to 5.0 and the rest to zero, the data for the two systems would look like the following:

Original System	A = 5.0, 9*0
Sigma 5	A=5.0,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,

As can be seen for large arrays the second method becomes quite tedious. The latest version of the Sigma 5 compiler is slightly larger than the one present when this work began. For this reason another modification had to be implemented to enable DREAD1 to compile. DREAD1 is a very large and complex subroutine which controls the actual reading and assembling of input data.

Because the new compiler was not able to handle the entire subroutine, DREAD1 was essentially divided in half. The first half of the input was processed by DREAD1 while the second half was controlled by the DREAD subroutine. NAMELIST variables had to be defined properly with the last section of cards in the DREAD1 subroutine switched to the end of the DREAD routine. This revision is simply a shifting of cards in order to reallocate space so that the program will fit into memory.

DPRINT,TPRINT: These two subroutines control the output for the system. DPRINT will print the input data along with the physical configuration for the problem. TPRINT controls the printout for the composition and conditions of all process streams. In the original version the user could specify a variable NPFREQ which permitted a printout of the intermediate results. This option has been deleted in the modified version due to problems in overlaying. However, the same effect can be obtained by decreasing the number of recycle loops for a problem run. After the specified number of loops the program will print out the final results. The problem can then be restarted from its present position by inserting a NOCLEAN card with no additional data. This, in effect, permits the user to obtain intermediate stream information even though the system has not converged. Also in the original CHESS program, DPRINT and TPRINT were two separate subroutines. Because of storage considerations they have been combined into

a single routine, decreasing the total amount of storage required. It will be seen later that this procedure has been applied to other modules.

DCHECK: This subroutine has been completely eliminated from the system. Its sole purpose was to do consistency and closure tests on the process matrix data. It is not needed if the data is properly entered and even if an error has occurred it can be found from an examination of the resulting printouts from DPRINT and TPRINT.

EQCALL: This routine calls the necessary equipment modules. Because of the nature of the overlay loader for the Sigma 5, individual call statements must be included for each segment. In other words, when an equipment module is desired, the segment containing that module must be called into core by means of a Fortran statement of the form CALL SEGLOD(I) where I is the desired segment (see discussion of program overlaying for more details). EQCALL is the only routine which references all the process modules. In some problems all of the modules are not needed and storage can be saved by not reading them onto the rad. In order to eliminate an unsatisfied reference error message in these cases, the call statements for these modules must be removed from EQCALL. These are special cases, however, and it will be assumed that the system is capable of calling all the process modules.

TEST: The TEST subprogram has been reduced substantially. It originally had two purposes. First, it checked for convergence of each of the streams during the recycle loop. If the condition of each stream equalled to the condition of that stream for the previous loop, execution was terminated and convergence was obtained. The second function of TEST was to determine new stream conditions by applying Wegstein's algorithm for convergence promotion. It was found that this convergence promotion did not always work well, causing oscillations in the results when a large number of components were present. This convergence promotion scheme was deleted from the Sigma 5 version, thus reducing the size of the TEST subroutine and the KE4 array.

Supporting Programs

All equipment modules and the thermodynamic package were left as they originally appeared with the exception of the dimensions of the common blocks which will be discussed next. These dimension changes made it necessary to change the limits on some of the DO loops in the supporting programs and some of the arguments in call statements. For example ABS00190 was changed from CALL ZERO (EQR,20) to CALL ZERO(EQR,10). Similar changes had to be made throughout the system. A number of changes had to be made due to the installation of the new compiler. The area designated as the FP area (rad area where programs are stored) was decreased by 16 files. This reduction of rad area was circumvented by

combining a number of subroutines, defining some as entry points instead of complete subroutines. The common blocks for each of the subroutines were combined causing a more efficient allocation of FP area. This combination of subprograms was applied to the following modules:

HXER and FHTR

ABSR and DVDR

MIXR and PUMP

These changes enabled all the necessary subroutines to fit into the available FP and SP area while not producing any problems as far as system configuration. Of course the selection of which programs that were to be combined was made on their position in the chosen overlay structure which will be discussed later. The thermodynamic package has not changed and its limitations remain the same as those specified in the manual.

Data Structures

Problem information for stream and equipment matrices in CHESS is stored in labeled common blocks and is available to all process modules. In order to fit the entire system on the Sigma 5, the original dimensions for the data structures were reduced by approximately a factor of two. Thus, the new

capacity for the system is limited to:

1. 50 process streams
2. 25 equipment nodes in the process network
3. 10 components
4. 4 streams at each equipment node (4 in/4 out)

Input Data Specifications

As was mentioned previously, the specifying of input data has been changed somewhat due to the nature of NAMELIST input on the Sigma 5 and also because of the decrease in the size of the data blocks. The data for an example problem will be presented later in order to show exactly how the data is to be specified. Although the same concept will be employed in the new version, there are a number of differences.

The first card read will be the same no matter what type of problem is to be executed. It will contain the words CLEAN, NAMELIST if a new problem is to be run with all data areas initialized to zero, or NOCLEAN, NAMELIST in which the data areas are preserved and the user may specify new equipment parameter vectors and/or stream data but not a new process configuration.

The second card must be a title card. It does not matter what is written on this card and its contents will be printed out as a title with all output associated with that particular run.

Following the title card are the data cards for the six NAMELIST groups. These groups, although not referred to by name

as in the normal Fortran compiler, must appear in the correct order and each is followed by an asterisk instead of the &END.

The changes in the input arrays are as follows:

PMLIST: The entry for the external equipment name in the KPM matrix has been eliminated reducing the size of the vector from 10 to 9. Also, the equipment module name which is the second entry of the KPMJ vector has been changed to a number code (see DREAD). The dimension on COMPNT has been decreased from 20 to 10 and the KOMNAM array has been decreased from 80 to 40.

The variable IPUNCH has been deleted entirely from this NAMELIST group.

EQLIST: The structure for the equipment input data has been unchanged but the maximum number of equipment nodes has become 25. Thus, ENAME is dimensioned 25 instead of 50 and there are only 25 EQPJ vectors.

SEXLIST: The total number of streams is limited to 50 causing the SNAME vector to be changed accordingly. Since the number of components available has been reduced to 10, the extensive property vectors SEXJ must also be reduced by 10. Thus, the vector is of size 13 with the same convention used as that in the original version.

SINLIST: Again, the SNAME vector must be decreased in dimension to 50 entries. Also, the intensive stream vector SINJ has been reduced by deleting its last four entries. Because viscosity,

thermal conductivity, and compressibility factors for the liquid and vapor phases are not supported in the CHESS thermodynamic package, they are omitted in the current system. The SINJ vectors are of length 6 with the following order of properties: stream numbers, stream flag, vapor fraction, temperature, pressure, and total heat content.

KELIST: The variable NPFREQ and KTRACE have been deleted as has been the KE4 matrix for convergence promotion. Also, the sizes of the KE2 and KE3 arrays have been reduced to 25.

NSCOMP: This NAMELIST group has been temporarily removed from the system because it was not needed in the problems to be run. However, if desired it could easily be placed back into the DREAD routine. Since the variables to be read are already assigned space in the common blocks, they would only have to be defined as NAMELIST variables and the corresponding read statements added.

Operation of the Sigma 5

Because this version of CHESS has been designed to run exclusively on the Sigma 5, it is important that the user be familiar with the basic operations of the computer software. In this way he will understand the reasons behind different procedures applied to CHESS and be able to modify or adapt for any future system changes. There are three primary operations that will be discussed here: reallocation of rad area, rad editing, and overlaying.

Reallocation of Rad Area

The Sigma 5 is a rad-oriented machine. This means that programs are read, compiled, and loaded onto the rad before they are transferred into core for execution. These three operations are especially critical when large programs are to be executed. The rad is subdivided into a number of different areas each designated to perform certain operations. The BT area is a type of I/O unit for the rad and is an integral part of every program run. The BT area is further subdivided into the GO file, the OV file, and X_1 files. The GO file is that area in which a program read in through the card reader is compiled and stored. It is a temporary storage location and holds the compiled program until the user tells the machine what to do with it. For small jobs the overlay loader takes the compiled program from the GO file and loads it into the OV file. The OV file is a core image and the user usually specifies that the machine run the OV file (!ROV) and the program will be read into core and executed. The other section of the BT area is the X_1 files which are used by the overlay loader in loading the programs for execution. The BT area is fixed in size by the system software and cannot be easily changed by the user. Likewise, the system software also allocates storage for each of the different files within the BT area.

However, the user may reallocate these files to suit his particular problem and this will be done for CHESS. Because the OV file must contain the entire loaded program, it must be made extremely large. Since the program will be overlaid to fit into core, the X_i files will in turn require a large area. Thus, very little space will remain for the GO file which will restrict the number of cards that can be read in through the card reader. The actual allocating of the individual files is a trial-and-error operation. All that the user is aware of is that the total area must equal the total size of the BT area. The usual number of files in these areas are:

GO = 192 files

OV = 192 files

X_i = 96 files

The reallocation of file sizes is accomplished with control cards immediately following the job card. The GO and OV files must be defined first and the six X_i files specified before job loading. The control cards for reallocation of the BT area for CHESS are given below:

```
!ALLOBT(FILE,GO),(FORMAT,B),(FSIZE,30),(RSIZE,30)
```

```
!ALLOBT(FILE,OV),(FSIZE,312)
```

```
!ALLOBT(FILE,X1),(FSIZE,0),SAVE
```

```
!ALLOBT(FILE,X2),(FSIZE,115)
```

```
!ALLOBT(FILE,X3),(FSIZE,10)  
!ALLOBT(FILE,X4),(FSIZE,10)  
!ALLOBT(FILE,X5),(FSIZE,10)  
!ALLOBT(FILE,X6),(FSIZE,0),SAVE
```

There are two other distinct areas of the rad, the FP area and the SP area. Both areas are designated as permanent storage locations. All system programs such as the Fortran compiler, overlay loader, system monitor, and system library are located in the SP area. In the FP area are stored user library routines such as the hybrid library. Because the GO file is limited restricting the number of cards that can be read, the CHESS subroutines have to be permanently stored in the machine. The FP and SP areas are the only places large enough to be able to handle the task. Similar to the BT area, the FP area has been assigned a finite number of files or storage locations by the system software. The hybrid routines occupy a small portion of this area, leaving enough room for the CHESS programs. Reading and storing programs into this area will be discussed in rad editing.

Rad Editing

Rad editing is the operation of reading programs onto the rad for later use or of deleting programs already stored there. In many cases efficient use of the Sigma 5 would require that rad editing be employed. For example, if every time a hybrid program

were to be run, all hybrid subroutines had to be read in through the card reader, much time would be wasted. Instead these routines are located in the FP area and when they are called from a source program, they are immediately available to the user program. Likewise, each segment of the CHESS program can be stored in the FP area, having to be read in only once. With the proper control cards the overlay loader is told to locate and retrieve the CHESS segments from that area when the program is loaded. Even though this is permanent storage, if a correction has to be made in a particular segment, that segment is deleted, the error corrected, and the segment placed back in the same location. Each segment, which may consist of one or more subroutines, is identified by a label or name as defined by the user. The rad editing procedure is quite simple. First, the area is usually mapped in order to find out how many programs are already stored there and how much space they require. This is accomplished with the control cards:

```
!JOB
!ATTEND
!RAEDIT
!TRUNCATE FP
!SQUEEZE FP
!MAP FP
!FIN
```

The first two cards tell the machine that an operation on the rad is going to follow. The next cards will cause the area to arrange itself in the most efficient manner. As the cards suggest, the programs residing there are squeezed in order to release as much space as possible for other programs. The programs are stored on the rad one behind the other listed from low file numbers to high file numbers. If a program in the lower files is deleted, there must be some provision for shifting all the other programs down to make more room at the high end. This is the function of the truncate and squeeze cards. The map card will cause the printing out of all programs located in that area with the file numbers on which they begin and end. This enables the user to determine how many files are available for other programs. The user after deciding that his program will fit into the FP area employs the following procedure to read his program into that area:

!JOB

!FORTRANH GO

FORTRAN DECK

!ATTEND

!RADEDIT

:ALLOT(FILE,FP,EXAMP),(FORMAT,B),(FSIZE,100),(RSIZE,30)

:COPY(FILE,BT,GO),(FILE,FP,EXAMP)

The first two cards and the source program are read into the GO file just as any other program that is to be run. The next two cards allow a rad operation to be performed (releases memory protect). The fifth control card will allot a file in the FP area called EXAMP that will reside within the space of 100 files. If the number of files requested by the user is not available in the FP area, a rad overflow message will occur. The copy card will transfer the contents of the GO file (source program) into the file labeled EXAMP. The subprogram is now located permanently in the FP area under the name EXAMP. A good technique when rad editing is to read the truncate, squeeze, and map cards after the program has been copied onto the rad. This releases any files that were allotted for EXAMP but not used. This will often occur because the FSIZE entry has to be greater to or equal to the number of files required by the program. If FSIZE is underspecified, an error message will result and the file will remain unchanged. In this case the file (EXAMP) has to be deleted and then reallocated with a larger file size. In order to delete a file, the card

```
:DELETE(FILE,FP,EXAMP)
```

along with the attend and radedit cards will delete program EXAMP. Again, the truncate, squeeze, and map should be run through to compact the area and recover those files previously allotted.

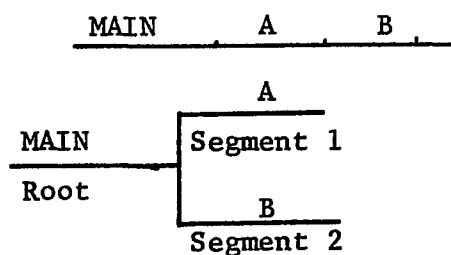
Overlaying

The only purpose of overlaying a program is to decrease the amount of core storage necessary for execution. Instead of reading an entire program into memory, an overlaid program is executed in parts called segments. The program is assembled or loaded on the rad by the overlay loader which connects the different segments together as specified by the user. The system software will then read the segments into core from the rad as they are called for by the root segment (or any other segment that is in core at that time). The root segment is that part of the program that is always located in core. Since the main program must remain in core at all times (there is no provision for calling the main program from any subroutine), it is always part of the root. However, the root might contain a number of other subroutines. Because it requires time to read segments into core from the rad, it would be desirable to have as few segments as possible. Also, as the number of segments increases, so does the number of X_i files used by the overlay loader. Thus, the ideal case would have only one segment which would be in core at all times. However, CHESS is too large for ordinary execution on the Sigma 5 and it must be segmented and executed in parts. The first step in overlaying a program is to draw a flow diagram or tree which contains all the subroutines and describes how they are interconnected. The appearance of this tree will be determined by the calling sequence of the program or how the subroutines call each other. This will dictate how the program

can be segmented. Consider the following example:

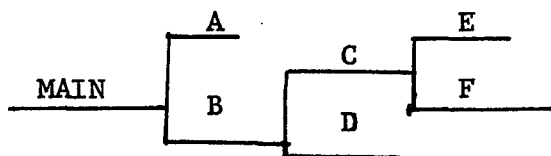
1. MAIN program calls subroutines A and B
2. A and B do not reference each other.

This program could be represented by the following trees:



In the first diagram all three subroutines are loaded into memory at one time. The lengths of the lines might represent the amount of memory required for each routine. In the second diagram, A and B are not in core storage at the same time and this configuration will require less core storage. However, because they are not in core at the same time they cannot call each other. Also, any data computed in a segment is destroyed when that segment is removed from core. Any information that is to be saved must be picked up through arguments of a call statement or in common blocks. In this example, the root would have to retrieve this data because it is the only segment present with both A and B. A flow diagram for CHESS to determine which subroutines called which and a tree was drawn (see Figure A3.1). There were numerous configurations possible and it had to be determined which was the most efficient. The program had to be divided to make as few segments as possible and still be able to fit into the

available core. It has to be remembered that the longest leg, that is the length from the root through the branches of the tree in the longest route, has to require less space than that available in core. For example, in the configuration



the subroutines MAIN, B, C, and F will all be present in memory at one time and they have to require fewer storage locations than that available in core. As explained previously, all common blocks were placed in the root which simplified data transfer. No special steps had to be taken to make sure that the data blocks were read in and out of core correctly. After a configuration was decided upon two types of control cards were needed. The first is used by the overlay loader and it tells the loader where to find a particular segment and then where to place it when loading. CHESS programs will be located in two areas. The root segment will be read in by means of the card reader and is found in the GO file. All other segments will be stored permanently in the FP and SP area of the rad. In summary, in a CHESS program only the root will have to be read in every time. The control cards for connecting or linking the segments are placed immediately after the OLOAD card. The first card

```
:ROOT(FILE,BT,GO,2)
```

tells the loader that the program root is found in the GO file of the BT area and it will consist of two subprograms. The root is automatically assigned as segment zero by the loader. The remainder of the control cards for loading are of the form:

SEG(LINK,20,ONTO,1),(FILE,FP,CHESS1,2)

This card instructs the loader to retrieve the two subroutines labeled CHESS1 located in the FP area and to define them as segment 20 and to link them onto segment 1. The numbers assigned to each segment are arbitrarily selected by the user after deciding on the system configuration. The numbers are the method by which the overlay loader keeps track of the different segments. Inspection of the complete list of control cards and program tree listed in the appendix will aid the user in understanding these machine instructions. Upon completion of loading, all CHESS programs will be linked together properly in the OV file and execution will begin. The other software command used in overlaying is a Fortran statement located in the source deck. The purpose of this command is to read the necessary segments into memory. This statement is of the form

CALL SEGL0D(I)

where I is the number of the segment needed in core. If the user has segmented and linked his program properly, he has only to insert these call statements into his source deck before execution of a subroutine call. In other words, if a program is executing and a subprogram is referenced that is located in a segment not residing in core, the call SEGLOD must be inserted to call that segment into core. If a reference is made to a segment already in core, another call SEGLOD is unnecessary. A useful tool in debugging and understanding overload programs is the program map. If requested, the machine will print out information on the size of all segments, how they are connected, how they are positioned in core, and all references made by each. The map is also invaluable in cases where there are numerous possibilities of how one should divide or segment a program. The purpose of any of the other control cards used in running the CHESS system on the Sigma 5 can be found in the reference manual for the machine.

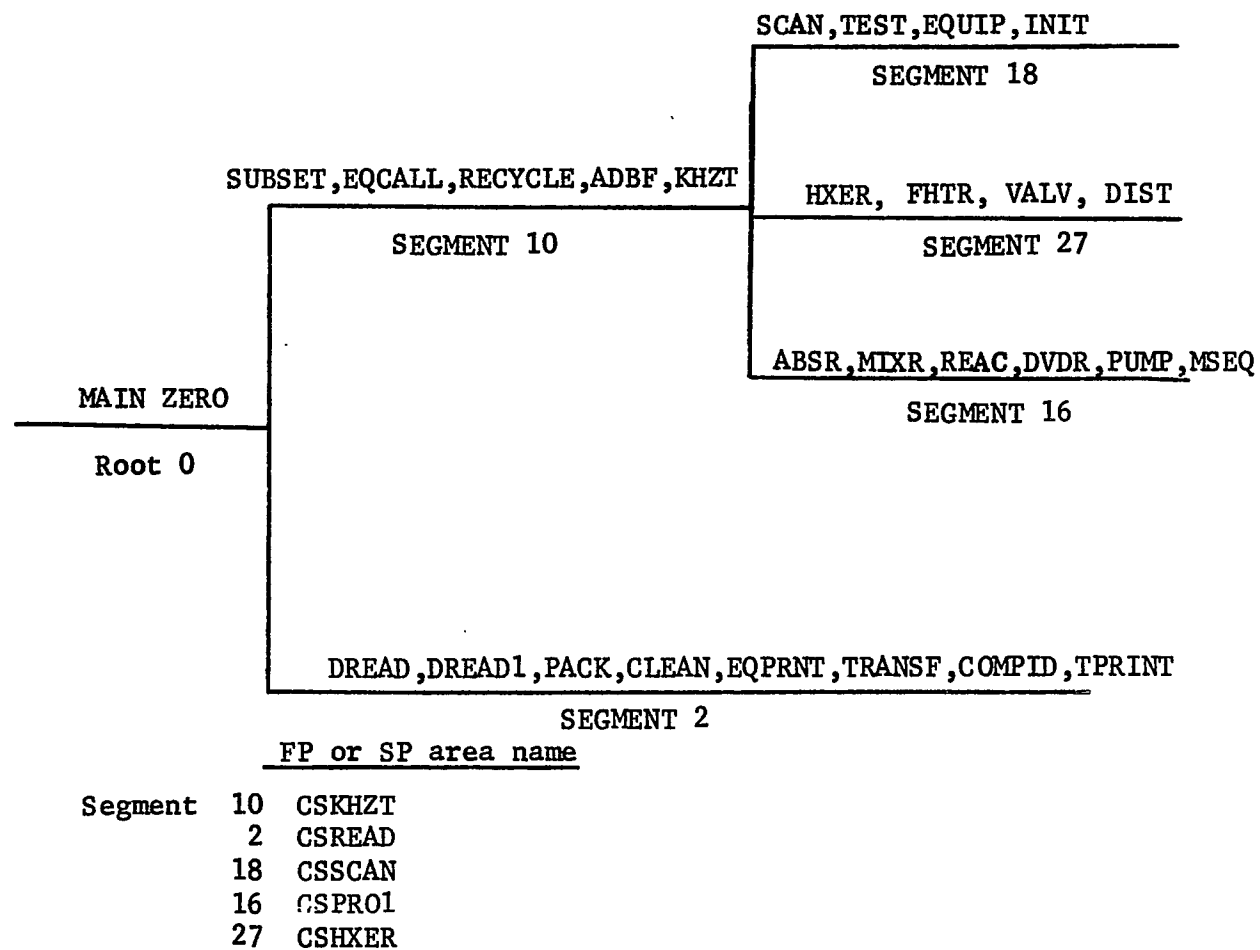


Figure A3.1. Overlay structure for CHESS on the Sigma 5.

APPENDIX D
COMPUTER PROGRAMS

C
C
C
C
C

PROGRAM THAT USES SENSITIVITY COEFFICIENTS
TO UPDATE MODEL AND RESPONSES FOR
SEVERAL DIFFERENT CONTROL TECHNIQUES

BLACK DATA

COMMON /POSCH/XCP,J,A3,VR,RH0B,TWIN,VT,RH0T,CP,W,DELHM,XKC,XA,
*CAIN,TR,NPT

DATA XCP/1.0/,U/1.25/,AB/200.0/,VB/8.64/,RH0B/62.4/,TWIN/80.0/,
1 VT/13.38/,RH0T/55.0/,CP/0.9/,W/73.5/,DELHM/12000.0/,
2 XKC/8.33E 08/,XA/14000.0/,CAIN/0.5975/,TR/205.31/

END

```

COMMON ZT(250),ZTP(250),ZXM(250)
COMMON /PU SCH/XCP,U,AB,VB,RHBT,TWIN,VT,RHBT,CP,W,DEI HM,XKC,YA,
*CAIN,TR,NPT
DIMENSION XAX(1),YAX1(3),YAX2(4)
DIMENSION BUF(1000)
DATA XAX/'TIME'//,YAX1(1)/'REAC'//,YAX1(2)/'TDR '//,YAX1(3)/'TEMP'//
DATA YAX2(1)/'WATE'//,YAX2(2)/'FL'//,YAX2(3)/'W R'//
DATA YAX2(4)/'ATE'//
CALL PLBTS(BUF,1000)
CALL PID
ZXM(NPT+1)=0.
ZXM(NPT+2)=20.
ZT(NPT+1)=0.
ZT(NPT+2)=2.
ZTP(NPT+1)=222.
ZTP(NPT+2)=2.
PRINT10
10  FORMAT(1H0,10X,'**PID CONTROLLER RESULTS**'//)
    PRINT13
13  FORMAT(2X,'NPT',6X,'TIME',6X,'TEMP',3X,'WATER RT')
    NPP=NPT+2
    DO2 I=1,NPP
2    PRINT1,I,ZT(I),ZTP(I),ZXM(I)
    CALL PLBT(0.,0.,-3)
    CALL AXIS(0.,0.,YAX1,12, 5.,90.,ZTP(NPT+1),ZTP(NPT+2))
    CALL AXIS( 0.,0.,XAX,-4, 5.,0.,ZT(NPT+1),ZT(NPT+2))
    CALL PLBT(0., 5.,3)
    CALL PLBT(5.,5.,2)
    CALL PLBT(5.,0.,2)
    CALL PLBT(0.,0.,3)
    CALL FLINE(ZT,ZTP,NPT,1,0,0)
    CALL PLBT(8.,0.,-3)
    CALL AXIS(0.,0.,YAX2,15, 5.,90.,ZXM(NPT+1),ZXM(NPT+2))
    CALL AXIS(0.,0.,XAX,-4, 5.,0.,ZT(NPT+1),ZT(NPT+2))

```

```

CALL PL9T(5.,0.,3)
CALL PL9T(5.,5.,2)
CALL PL9T(0.,0.,3)
CALL FLINE(ZT,ZXM,NPT,1,0,0)
CALL PL9T(-8.,0.,-3)
CALL KALMAN
D05 I=2,NPT
ZTP(I-1)=ZTP(I)
ZXM(I-1)=ZXM(I)
CONTINUE
NPT=NPT-1
ZXM(NPT+1)=0.
ZXM(NPT+2)=20.
ZT(NPT+1)=0.
ZT(NPT+2)=2.
ZTP(NPT+1)=222.
ZTP(NPT+2)=2.
PRINT 11
FORMAT(1H0,10X,'**KALMAN CONTRALLER RESULTS**'/)
PRINT13
NPP=NPT+2
D03 I=1,NPP
PRINT1,I,ZT(I),ZTP(I),ZXM(I)
CALL FLINE(ZT,ZTP,NPT,1,0,0)
CALL PL9T(8.,0.,-3)
CALL FLINE(ZT,ZXM,NPT,1,1,4)
CALL PL9T(-8.,0.,-3)
CALL BST
ZT(NPT+1)=0.
ZT(NPT+2)=2.
ZTP(NPT+1)=222.
ZTP(NPT+2)=2.
ZXM(NPT+1)=0.
ZXM(NPT+2)=20.

```

5

11

3

```

PRINT12
12  FORMAT(1H0,10X,'**SENSITIVITY COEFFICIENTS AND 9ST')
PRINT13
NPP=NPT+2
DO4 I=1,NPP
4  PRINT1,I,ZT(I),ZTP(I),ZXM(I)
1  FORMAT(1X,I5,3F10.3)
CALL FLINE(ZT,ZTP,NPT,1,0,0)
CALL PLOT(8.,0.,-3)
CALL LINE(ZT,ZXM,NPT,1,0,0)
CALL PLOT( 8.,0.,-3)
CALL PLOT(0.,0.,999)
STOP
END

```

```

C
C
C
C
SUBROUTINE PID
  USING OPTIMAL TUNING CONSTANTS AND P-I CONTRAILER
  WITH CONSTRAINTS

  COMMON ZT(250),ZTP(250),ZXW(250)
  COMMON /PUSCH/XCP,U,AB,VR,RH0B,TWIN,VT,RH0T,CP,W,DEIHM,XKC,XA,
  *CAIN,TR,NPT
  DIMENSION P(3),STEP(3)
  P(1)=12.2
  P(2)=48.7
  SETPT=225.
  TW=209.76
  TP=230.
  CA=0.1802
  XMSS=38.995
  DERE=0.
  DEREL=0.
  E=0.
  ELAST=0.
  C0ST=0.
  XM=XMSS
  H=0.001
  T=0.
  IPRINT=1
  JPRINT=60
  PRINT5
5  F0RMAT(12X,'TIME',3X,'REACTOR TFP',10X,'ERROR',5X,'WATER RATE',
  112X,'COST'/)
  NPT=1
  ZT(NPT)=T
  ZTP(NPT)=TP
  ZXW(NPT)=100.
1  CONTINUE

```

```

T=T+H
IPRINT=IPRINT+1
XK=XKC*EXP(-XA/(TP+460.))
TW=(XM*XCP*(TWIN-TW)+U*AB*(TP-TW))/(VB*RH03*XCP)
TW=TW+TW*H
TPF=(W*CP*(TR-TP)+DELHM*XK*CA*CA*VT-U*AB*(TP-TW))/(VT*RH0T*CP)
TP=TP+TPF*H
CAF=(W*(CAIN-CA)-RH0T*VT*XK*CA*CA)/RH0T/VT
CA=CA+CAF*H
E=TP-SETPT
DERE=(E-ELAST)/4
DELM=P(1)*(DERE+1./P(2)*E)
XM=XM+H*DELM
IF(XM.LT.0.)XM=0.
IF(XM.GT.100.)XM=100.
DEREL=DERE
ELAST=E
COST=COST+H*T*ABS(E)
IF(IPRINT.EQ.JPRINT)GOTO11
GOTO12
11 CONTINUE
NPT=NPT+1
ZT(NPT)=T
ZTP(NPT)=TP
ZX(NPT)=XM
PRINT3,T,TP,F,XM,COST
FORMAT(1X,5F15.4)
IPRINT=1
12 CONTINUE
IF(T.LE.10.)GOTO1
RETURN
END

```

```

C
C
C
SUBROUTINE KALMAN
  USING KALMAN CONTRROLLER WITH CONSTRAINTS ON OUTPUT

COMMON ZT(250),ZTP(250),ZXM(250)
COMMON /PUSCH/XCP,J,A3,VB,RHB3,TWIN,VT,RHBT,CP,W,DEIHM,XKC,XA,
*CAIN,TR,NPT
27 READ25,SAMT,XFIN
25 FORMAT(2F10.2)
PRINT26,SAMT
26 FORMAT(1H1,5X,'SAMPLING TIME',F10.4///)
SETPT=225.
TW=209.76
TP=230.
CA=0.1802
XMSS=38.995
C1=0.72588
C2=0.05131
C3=-0.08648
GAIN=C3/C2
TAU1=(C1+SQRT(C1*C1+4.*C2))/2./C2
TAU2=1./C2/TAU1
CBST=0.
XM=XMSS
H=0.001
T=0.
JPRINT=SAMT/H+1.
DTAU=TAU2-TAU1
ET1=EXP(-SAMT/TAU1)
ET2=EXP(-SAMT/TAU2)

  CALCULATION OF KALMAN CONSTANTS

  A1=GAIN/DTAU*(DTAU*(-ET1-ET2)-TAU1*(1.+ET2)+TAU2*(1.+ET1))
C
C
C

```

```

20      A2=3*IN/DTAU*(DTAU*ET1*ET2+TAU1*ET2+TAU2*ET1)
      PRINT20,A1,A2,TAU1,TAU2
      FORMAT(1X,'ORIGINAL P CREF',2F10.4//1X,' TIME CONSTANTS',2F10.4//)

C
C
C      NORMALIZING KALMAN CONSTANTS

      XN=1./(A1+A2)
      B0=XN
      B1=XN*(ET1+ET2)
      B2=XN*(ET1*ET2)
      A1=A1*XN
      A2=A2*XN
      PRINT21,XN,B0,B1,B2,A1,A2
      FORMAT(1X,'SCALE FACTOR',F10.4//1X,' COEFFICIENTS',3F10.4//
      11X,' P COEFFICIENTS',2F10.4//)
      XMN1=0.
      XMN2=0.
      E1=0.
      E2=0.
      IPRINT=1
      PRINT5
      FORMAT(12X,'TIME',3X,'REACTOR TEMP',10X,'ERROR',5X,'WATER RATE',
      112X,'CAST'//)
      NPT=1
      ZT(NPT)=T
      ZTP(NPT)=TP
      ZX(NPT)=XM
      CONTINUE
      T=T+H
      IPRINT=IPRINT+1
      XK=XKC*EXP(-XA/(TP+.60.))
      TW=(XM*XCP*(TWIN-TN)+U*AB*(TP-TN))/(VR*RH03*SCP)
      TW=TW+TW*H
      TPF=(W*CP*(TR-TP)+DEHN*XK*CA*VT-U*AB*(TP-TN))/(VT*RH0T*CP)

```



```

TP=TP+TPF*H
CAF=(J*(CAIN-CA)-RHUT*VT*XK*CA*CA)/RHUT/VT
CA=CA+CAF*H
E=SETPT-TP
COST=COST+H*T*A3S(F)
  IF(IPRINT.EQ.JPRINT)GOTO11
GOTO12
CONTINUE
PRINT3,T,TP,E,XM,COST
FORMAT(1X,5F15.4)
NPT=NPT+1
ZT(NPT)=T
ZTP(NPT)=TP
IPRINT=1
XMN=BO*E+B1*E1+32*E2+A1*XMN1+A2*XMN2
XM=XMN+XMSS
  IF(XM.LT.0.)XM=0.
  IF(XM.GT.100.)XM=100.
XMN=XM-XMSS
E=(XMN-(B1*E1+B2*E2+A1*XMN1+A2*XMN2))/BO
XMN2=XMN1
XMN1=XMN
E2=E1
E1=E
ZXM(NPT)=XM
CONTINUE
  IF(T.LE.10.)GOTO1
  IF(XFIN.LT.0.5)GOTO27
RETURN
END

```

11

3

12

```

SUBROUTINE BST
C
C      USE OF SENSITIVITY COEFFICIENTS TO UPDATE MODEL PARAMETERS
C
C      USE OPTIMAL SWITCHING TIMES FOR BANG-BANG CONTROL
C
COMMON ZT(250),ZTP(250),ZXM(250)
COMMON /PUSCH/XCP,J,AB,VB,RH0B,TWIN,VT,RH0T,CP,4,DEIHM,XKC,XA,
*CAIN,TR,NPT
DIMENSION AMAT(3,3),RMAT(3)
DIMENSION DQ(3)
A=0.7
B=0.05
GAIN=0.1
T1=2.76
T2=3.48
TP=230.
TW=209.76
CA=0.18
XMSS=38.995
READ30,R,IFIN
FORMAT(F10.6,I10)
TPSS=TP+R
XMIN=XMSS
CALL SSTATE(TPSS,CASS,TWSS,XMSS)
TPDUM=TP
CADUM=CA
TWDUM=TW
PRINT31
FORMAT(1H1,1X,'PRESENT AND DESIRED CONDITIONS',/)
PRINT115
PRINT115,TIME,TP,XMIN,CA,TW
PRINT115,TIME,TPSS,XMSS,CASS,TWSS
DO900 II=1,3

```

44
30

31

```

TP=TPDUM
CA=CADUM
TW=TWUDUM
TPIN=TP
D0199 I=1,3
DQ(I)=0.
BMA(I)=0.
D0199 J=1,3
AMAT(I,J)=0.
D1=100.
D2=0.
RK=D1
SK=D2
IF(R.GT.0.)SK=D1
IF(R.GT.0.)BK=D2
TIME=0.
H=0.001
COST=0.
COST2=0.
C=0.
C1=0.
ZUA=0.
ZUR=0.
ZUK=0.
UA=0.
UB=0.
UK=0.
IPRINT=1
116 FORMAT(1H0,17X,'TIME',8X,'REACTOR TEMP',10X,'WATER RATE',7X,
2'CNCENTRATION',9X,'JACKET TEMP'//)
119 FORMAT(1H0,17X,'TIME',8X,'REACTOR TEMP',10X,'MODEL TEMP',7X,
5'WATER FLOW RT',9X,'MODEL FRRR'//)
PRINT119
NPT=1

```

```

ZTP(NPT)=TP
ZXX(NPT)=100.
7T(NPT)=TIME
CONTINUE
IF(TIME.LT.T1)GOTO1
IF(TIME.LT.T2)GOTO2
XM=XMSS
GOTO3
1  XM=BK
2  GOTO3
3  XM=SK
CONTINUE
IPRINT=IPRINT+1
TIME=TIME+H
XK=XKC*EXP(-XA/(TP+460.))
TW=(XM*XCP*(TWIN=TW)+U*AB*(TP-TW))/(VB*RHOB*XCP)
TW=TW+TW*H
TPF=(W*CP*(TR-TP)+DEL-HM*XK*CA*CA*VT-U*AB*(TP-TW))/(VT*RHOB*CP)
TP=TP+TPF*H
CAF=(W*(CAIN=CA)-RHBT*VT*XK*CA*CA)/RHBT/VT
CA=CA+CAF*H
FT=XM-XMIN
FORCE=FT
C1F=GAIN*FORCE-A*C1-R*C
C1=C1+H*C1F
C=C+H*C1
CDIFF=TP-TPIN=C
SC9ST=TIME*ABS(CDIFF)
C9ST=C9ST+H*SC9ST
C9ST2=C9ST2+H*TIME*ABS(TP-TPSS)

C
C SENSITIVITY COEFFICIENTS
C JA,UB,UK WITH DERIVATIVES ZUA,ZUB,ZUK
C

```

```

ZJAF=-A*ZUA-C1-B*UA
ZUBF=-A*ZUB-B*UB-C
ZUKF=-A*ZUK-B*UK
ZJA=ZJA+H*ZJAF
ZUB=ZUB+H*ZUBF
ZUK=ZUK+H*ZUKF
UA=UA+H*ZUA
UB=UB+H*ZUB
UK=UK+H*ZUK
AMAT(1,1)=AMAT(1,1)+H*UA*UA
AMAT(1,2)=AMAT(1,2)+H*UA*UR
AMAT(1,3)=AMAT(1,3)+H*UA*UK
BMAT(1)=BMAT(1)+H*UA*CDIFF
AMAT(2,1)=AMAT(1,2)
AMAT(2,2)=AMAT(2,2)+H*UB*UR
AMAT(2,3)=AMAT(2,3)+H*UB*UK
BMAT(2)=BMAT(2)+UB*CDIFF*H
AMAT(3,1)=AMAT(1,3)
AMAT(3,2)=AMAT(2,3)
AMAT(3,3)=AMAT(3,3)+H*UK*UK
BMAT(3)=BMAT(3)+H*UK*CDIFF
IF(IPRINT.EQ.101)G9TA120
G9TA121
120  CONTINUE
Z=TPIN+C
PRINT115,TIME,TP,Z,XM,CDIFF
115  FORMAT(1X,5F20.4)
NPT=NPT+1
ZT(NPT)=TIME
ZTP(NPT)=TP
ZXM(NPT)=XM
IPRINT=1
121  CONTINUE
IF(TIME.LE.10.)G9TA114

```

```

58 PRINT58
   FORMAT(1H0,10X,'SENSITIVITY COEFFICIENT MATRIX')
   DO9C1 J=1,3
901 PRINT902,AMAT(J,1),AMAT(J,2),AMAT(J,3),BMAT(J)
902   FORMAT(10X,3F15.3,F15.3)
   CALL GAUSS(AMAT,BMAT,DQ)
   PRINT59
59   FORMAT(1H0,10X,'PARAMETER CHANGES')
   PRINT111,DQ(1),DQ(2),DQ(3)
111   FORMAT(11X,3F20.4)
   A=A+DQ(1)
   B=B+DQ(2)
   GAIN=GAIN+DQ(3)
   PRINT48,A,B,GAIN
48   FORMAT(1H0,5X,'MODEL CONSTANTS',3F15.5//)
   PRINT179,COST,CST2
179   FORMAT(1H0,1X,'MODEL COST',F10.4,5X,'SET PRINT COST',F10.4//)
   XKP=GAIN/B
   TAU=(A+SQRT(A*A+.4*B))/2./B
   SB=1./B/TAU/TAU
   PRINT24,TAU,SB,XKP
24   FORMAT(1H0,1X,'TAU=',F10.4//1X,'B=',F10.4//1X,'GAIN=',F10.4//)
   CALL QBTIME(TAU,SB,XKP,R,XMIN,T1,T2)
   PRINT28,T1,T2
28   FORMAT(1X//10X,'SWITCHING TIMEs',2F20.5//)
900 CONTINUE
   IF(IFIN.LT.1)GOTO844
   RETURN
   END

```

```

SUBROUTINE GAUSS(CI,R,DQ)
DIMENSION CI(3,3),B(3),DQ(3)
N=3
NN=N-1
D05I=1,NN
J=N+1-I
L=J-1
D06K=1,L
FACT=CI(K,J)/CI(J,J)
D07M=1,J
7 CI(K,M)=CI(K,M)-FACT*CI(J,M)
6 B(K)=B(K)-FACT*B(J)
5 CONTINUE
DQ(1)=B(1)/CI(1,1)
D011I=2,N
SUM=B(I)
M=I-1
D09J=1,M
9 SUM=SUM-CI(I,J)*DQ(J)
11 DQ(I)=SUM/CI(I,I)
RETURN
END

```

```

SUBROUTINE SSTATE(TP,CASS,TWSS,XMSS)
COMMON /PUSCH/XCP,J,A3,VR,RHBT,TWIN,VT,RHAT,CP,V,DELHM,XKC,YA,
*CAIN,TR,NPT
C
C   GIVEN REACTOR TEMPERATURE--SOLVE FOR STEADY STATE VALUES
C
XK=XKC*EXP(-XA/(TP+460.))
CASS=(-1+SQRT(W*W+4.*RHBT*VT*XK*W*CAIN))/(2.*RHBT*VT*XK)
TWSS=(U*A3*TP+W*CP*(TP-TR)-DELHM*XK*CASS*CASS*VT)/U/AB
XMSS=U*AB*(TWSS-TP)/XCP/(TWIN-TWSS)
RETURN
END

```

C
C
C


```

SUBROUTINE OPTIME(TAU,B,GAIN,R,X,T1,T2)
  XMSS=XM
  GP=GAIN
  XL=0.001
  XH=20.
  TOL=0.001
  RK=100.-X4
  SK=-XM
  IF(R.LT.0.)BK=-XM
  IF(R.LT.0.)SK=100.-XM
  XM=(XL+XH)/2.
  FXL=((BK-SK+SK*EXP(-XL/B/TAU))/(BK-R/GP))*3-(BK-SK+SK*EXP(-XL/TAU
  3))/((BK-R/GP)
  FXM=((BK-SK+SK*EXP(-XM/B/TAU))/(BK-R/GP))*3-(BK-SK+SK*EXP(-XM/TAU
  4))/((BK-R/GP)
  PRD=FXM*FXL
  IF(PRD.LT.0.)GOTO1
  XL=XM
  GOTO2
  XH=XM
  DIFF=XH-XL
  IF(DIFF.LT.TOL)GOTO3
  GOTO4
  CONTINUE
  T1=XL
  T2=TAU*ALAG((-SK-(BK-SK)*EXP(T1/TAU))/(R/GP-BK))
  XM=XMSS
  RETURN
  END

```

4

1

2

3

```

C
C PROGRAM TO PLOT RESPONSES OF CHEMICAL REACTOR
C FOR SEVERAL CONTROL SCHEMES
C
  DIMENSION BUF(1000)
  DIMENSION BRD(3),AA(1)
  DIMENSION PT(2000),PTP(2000)
  DIMENSION V(6),S(6),PAR(3)
  DIMENSION C1(3,3),C2(3,3),C3(3,3),C4(3,3),C5(3,3)
  DIMENSION G(3,3),A(3,3),B(3),R(3,3),AT(3,3),FR(3,3)
  DATA CP/0.9/,TR/150./,DELH/-867./,VT/250./,Z/3./,TWI/80./
  DATA XMCW/4000./,AB/200./,RHOT/60./,CA0/9./,ZKC/1.433/,Y/2560./
  DATA BRD/'TEMP','ERAT','URE'//,AA/'TIME'/
  CALL ERRSET(207,900,5,1)
  CALL PLOTS(BUF,1000)
  CALL FACTOR(0.8)
C ENTER CONSTANTS OF PROCESS MODEL PREVIOUSLY CALCULATED
  V(1)=-0.009555
  V(2)=12.603
  V(3)=2.5194
C
C ITER #1      SP      PID
C ITER #2      SP      PID
C ITER #3      LOAD    PID
C ITER #4      LOAD    PID
C
81  CONTINUE
  READ1,ITER,STIME,XITER,P,XDAVIS
  1  FORMAT(I1,F9.2,3F10.2)
C CALCULATE TUNING PARAMETERS FOR A GIVEN P - VALUE
  TAUUV=0.1
  TAUT=1.
  TP=190.
  TPM9=TP

```

```

TPVL=TP
CA=3.6
TW=120.
XMSS=1050.
ELAST=0.
EEL=0.
SX1=0.
TPSS=TP
COST=0.
XMLAST=XMSS
XMOLD=XMSS
TOUT=100.
H=0.01
CSET=190.
W=1000.
U=0.
IF(ITER.EQ.1.0R.ITER.EQ.2)U=500.
IF(ITER.EQ.1.0R.ITER.EQ.2)CSET=185.
IF(ITER.EQ.3.0R.ITER.EQ.4)W=1200.
T=0.
NPT=1
PT(NPT)=T
PTP(NPT)=TP
DUM=0.
SAMP=20.
T=T+H
SAMP=SAMP+H
DUM=DUM+C.1
TPM9=TPVL
CONTINUE
XMNEW=XMSS+U
XMF=(XMNEW-XMOLD)/TAUV
XM=XMOLD+H*XMF
XMOLD=XM

```

C

7

22

88

```

TWF=(XM*(TWI-TV)+Z*AB*(TP-TV))/XMCJ
TW=TW+H*TF
XK=ZKC*EXP(-Y/(TP+460.0))
CAF=W/VT/RHQT*(CAF*CA)-XK*CA*CA
CA=CA+H*CAF
TPF=(N*CP*(TR-TP)-DELH*XK*CA*CA*VT-Z*AB*(TP-TV))/VT/RHQT/CO
TP=TP+H*TPF
TPMN=TP
TPFF=(TPMN-TPML)/TAUT
TPM=TPML+H*TPFF
TPML=TPM
XMLAST=XM
IF(DUM.LT.1.)GOTO23
NPT=NPT+1
DUM=0.
PT(NPT)=T
PTP(NPT)=TP
CONTINUE
IF(ITER.EQ.3.6R.ITER.EQ.4)GOTO29
IF(T.GT.50.)U=0.
IF(T.GT.50.)CSET=190.
GOTO28
IF(T.GT.50.)W=1000.
IF(T.LT.TAUT)GOTO22
D684 J=1,NPT,10
PRINT21,PT(J),PTP(J),XM .CAST,J
FORMAT(1X,4F20.4,110)
FORMAT(1X,5F20.2)
PT(NPT+1)=0.
PT(NPT+2)=20.
PTP(NPT+1)=185.
PTP(NPT+2)=2.
CALL AXIS(0.,0.,90.,+11,5.,90.,PTP(NPT+1),PTP(NPT+2))
CALL AXIS(0.,0.,45.,-4,5.,0.,PT(NPT+1),PT(NPT+2))

```

23

C

29

28

84

21

20

```
CALL PLBT(0.,5.,+3)
CALL PLBT(5.,5.,+2)
CALL PLBT(5.,0.,+2)
CALL PLBT(0.,0.,+3)
CALL FLINE(PT,PTP,NPT,1,0,0)
CALL PLBT(8.,0.,-3)
IF(XDAVIS.LT.0.5)G9TA81
CALL PLBT(0.,0.,999)
STOP
END
```

```

C
C PROGRAM TO CALCULATE CONTROLLER CONSTANTS
C WITH MATRIX RICATTI EQUATION
C
C P'=P*K**2*TAU**2
C DIMENSIONLESS TIME
C NEGATIVE FEEDBACK
C
C DIMENSION Q(3,3),A(3,3),B(3),R(3,3),AT(3,3)
C DIMENSION C1(3,3),C2(3,3),C3(3,3),C4(3,3),C5(3,3)
C DIMENSION FR(3,3),YU(150)
C DIMENSION PT(500),PX1(500)
C DIMENSION XKCK(500),XTIOT(500),XTDAT(500),TGTAU(500)
C DIMENSION BUF(5000)
C DIMENSION BRD(2),AB(2)
C DIMENSION AB2(3),BRD2(1),BRD3(2),BRD4(2)
C DATA BRD/'RESPONSE'/,AB/'TIME/TAU'/
C DATA AB2/'THETA/TAU'/,BRD2/'KKc'/,BRD3/'TI/TAU'/,BRD4/'TD/TAU'/
C
C INTEGRATION OF MATRIX RICATTI EQUATION
C
C CALL ERRSET(207,900,5,1)
C TAU=1.
C GAIN=1.
C DIST=1.
C ADUM=0.
C XLAST=0.
C DAXIS=0.
C CLAST=0.
C SK=0.
C RK=2.
C STI=0.5
C BTI=1.75
C STD=0.

```

```

      BTD=0.5
      X1MIN=-0.25
      X1MAX=1.
98    CONTINUE
61    READ33,P,XFIN
33    FORMAT(2F10.2)
      THETA=0.05
      CLAST=1.
      NPT=0
63    CONTINUE
      NPT=NPT+1
      THETA=THETA+0.05
      RATIO=THETA/TAU
64    CONTINUE
      D02I=1,3
      B(I)=0.
      D02J=1,3
      C4(I,J)=0.
      Q(I,J)=0.
2     A(I,J)=0.
      Q(1,1)=1.
      B(3)=1.
      C8=THETA/2./TAU
      A(1,2)=1.
      A(1,3)=-1.
      A(2,1)=-1./C8
      A(2,2)=-(C8+1.)/C8
      A(2,3)=(2.+C8)/C8
      N=3
      M=1
      D04I=1,3
      D04J=1,3
      R(I,J)=0.
4     AT(I,J)=A(J,I)

```

```

      L=0.05
      TOUT=60.
      T=0.
      T=T+H
      CALL MTXMPY(AT,P,C1,N,N,N)
      CALL MTXMPY(R,A,C2,N,N,N)
      DO8I=1,3
      C4(I,3)=R(I,3)
      CALL MTXMPY(C4,R,C5,N,N,N)
      DO6I=1,3
      DO6J=1,3
      FR(I,J)=-C1(I,J)-C2(I,J)+C5(I,J)/P-G(I,J)
      R(I,J)=R(I,J)-H*FR(I,J)
      CONTINUE
      IF(T.LT.TOUT)GOTO11
      XK1=-R(1,3)/P
      XK2=-R(2,3)/P
      XK3=-R(3,3)/P
      XFACT=1./(1.-XK2*C8-XK3*C8)
      XFACT=XFACT
      Z1=XFACT*(XK1+XK2+XK3)
      Z2=XFACT*(XK2+(XK2+XK3)*(1.+C8))
      Z3=XFACT*(XK2*C8+XK3*C8)
      XKCK(NPT)=Z2/GATN
      XT10T(NPT)=Z2/Z1
      XTD9T(NPT)=Z3/Z2
      T0TAU(NPT)=RAT10
      IF(XKCK(NPT).LT.SK)XKCK(NPT)=SK
      IF(XKCK(NPT).GT.BK)XKCK(NPT)=BK
      IF(XT10T(NPT).LT.ST1)XT10T(NPT)=ST1
      IF(XT10T(NPT).GT.BT1)XT10T(NPT)=BT1
      IF(XTD9T(NPT).LT.ST0)XTD9T(NPT)=ST0
      IF(XTD9T(NPT).GT.BT0)XTD9T(NPT)=BT0
      PRINT25,XKCK(NPT),XT10T(NPT),XTD9T(NPT),T0TAU(NPT)

```

11

8

6


```

25  FORMAT(1X,'CONTROLLER CONSTANT',3F20.4,5X,'T=ETA/TAU='F10.2)
   IF(RATIO.LT.0.94)GOTO63
   PRINT84,P
84  FORMAT(1X,'***** P=1,F6.1)
   IF(DAXIS.GT.0.5)GOTO69
   XKCK(NPT+1)=SK
   XKCK(NPT+2)=(BK-SK)/5.
   XTIST(NPT+1)=STI
   XTIST(NPT+2)=(BTI-STI)/5.
   XTDST(NPT+1)=STD
   XTDST(NPT+2)=(STD-STD)/5.
   TETAU(NPT+1)=0.
   TETAU(NPT+2)=0.2
   CALL PLOTS(BUF,5000)
   CALL FACTOR(0.8)
   CALL PLOT(0,0.,-3)
   CALL AXIS(0,0.,90,2,+3,5.,90.,XKCK(NPT+1),XKCK(NPT+2))
   CALL AXIS(0,0.,90,2,-9,5.,0.,TETAU(NPT+1),TETAU(NPT+2))
   CALL PLOT(0,5.,+3)
   CALL PLOT(5,5.,+2)
   CALL PLOT(5,0.,+2)
   CALL AXIS(8,0.,90,3,+6,5.,90.,XTIST(NPT+1),XTIST(NPT+2))
   CALL AXIS(8,0.,90,3,-9,5.,0.,TETAU(NPT+1),TETAU(NPT+2))
   CALL PLOT(8,5.,+3)
   CALL PLOT(13,5.,+2)
   CALL PLOT(13,0.,+2)
   CALL AXIS(16,0.,90,4,+6,5.,90.,XTIST(NPT+1),XTIST(NPT+2))
   CALL AXIS(16,0.,90,4,-9,5.,0.,TETAU(NPT+1),TETAU(NPT+2))
   CALL PLOT(16,5.,+3)
   CALL PLOT(21,5.,+2)
   CALL PLOT(21,0.,+2)
   DAXIS=1.0
69  CALL PLOT(0,0.,-3)
   CALL FLINE(TETAU,XKCK,NPT,1,0,0)

```

```

      CALL PLBT(8.,0.,-3)
      CALL FLINE(TBTAT,XTIAT,NPT,1,0.0)
      CALL PLBT(8.,0.,-3)
      CALL FLINE(TBTAT,XTDAT,NPT,1,0.0)
      CALL PLBT(-16.,0.,-3)
      IF(XFIN.GT.0.5)GOTO988
      GOTO61
88    CONTINUE
      CALL PLBT(25.,0.,-3)
      CALL PLBT(0.,0.,999)
89    FORMAT(1X,2F20.4)
90    STOP
      END

```

```
SUBROUTINE MTXMDY(A,B,C,N,K,M)
  DIMENSION A(3,3), B(3,3),C(3,3)
  DO5I=1,N
  DO5J=1,M
  C(I,J)=0.
  DO5L=1,K
  C(I,J)=C(I,J)+A(I,L)*B(L,J)
  RETURN
  END
```

```

C
C PROGRAM TO DETERMINE RESPONSES OF
C MULTIVARIABLE SYSTEM FOR SEVERAL CONTROL SCHEMES
C
  DIMENSION DDYA1(60),DDYA2(60),DDYD1(60),DDYD2(60)
  DIMENSION XPTS(500),YPTS(500),YYPTS(500),ZPTS(500),ZZPTS(500)
  DIMENSION BUF(1000)
  DATA XAXIS/'TIME'/,YAXIS/'Y1'/,YYAXIS/'Y2'/,ZAXIS/'F'/,ZZAXIS/'T'/
  DATA TAUAF/2.0/,TAUDF/ 2.0/,TAHAT/ 4.0/,TAUDT/4.0/,YASS/2.005E-05/
  DATA YDSS/0.43500/,FSS/18.0/,TSS/800./,XKAF/2.5E-08/
  DATA XKAT/-2.5E-09/,XKDF/2.05E-02/,XKDT/-2.23E-03/
  DATA DEADAF/0.1/,DEADDF/0.1/,DEADAT/0.2/,DEADDT/0.2/
  DATA RGYAF/12.4/,RGYAT/-11.4/,RGYDF/-11.4/,RGYDT/12.4/

C
C ITYPE = 1 EQUALS Y1 CONTROLLED WITH FLOW
C ITYPE = 2 EQUALS Y2 CONTROLLED WITH TEMP
C ITYPE = 3 EQUALS BOTH LOOPS CONTROLLED
C ITYPE = 4 EQUALS CHANGE IN FLOW VARIABLE INTO SS DECOUPLER
C ITYPE = 5 EQUALS CHANGE IN TEMP VARIABLE INTO SS DECOUPLER
C ITYPE = 6 EQUALS CHANGE IN FLOW WITH DYNAMIC COUPLER
C ITYPE = 7 EQUALS CHANGE IN TEMP WITH DYNAMIC COUPLER
C ITYPE = 8 EQUALS Y1 CONTROLLED WITH FLOW WITH DYNAMIC DECOUPLER
C ITYPE = 9 EQUALS Y2 CONTROLLED WITH TEMP WITH DYNAMIC DECOUPLER
C ITYPE =10 BOTH LOOPS CONTROLLED WITH DYNAMIC DECOUPLER
C
  CALL PLSTS(BUF,1000)
  CALL FACTOR(0.6)
  REFYD=0.3
  REFYA=.199E-04
  Y1MIN=.198E-04
  Y1MAX=.203E-04
  Y2MIN=.25
  Y2MAX=.5
  FMIN=0.

```

```

      FMAX=100.
      TMIN=0.
      TMAX=1500.
      CONTINUE
30    DB100 ITYPE=1,1)
      DETERMINE DECOUPLER DENOMINATOR
      DET=XKAF*XKDT-X<AT*XKDF
      D640 K=1,60
      DDYA1(K)=0.
      DDYA2(K)=0.
      DDYD1(K)=0.
      DDYD2(K)=0.
40    H=0.01
      NYA1=DEADAF/H+.00001
      NYA2=DEADAT/H+.00001
      NYD2=DEADDT/H+.00001
      NYD1=DEADDF/H+.00001
      DYA1=0.
      DYA2=0.
      DYD1=0.
      DYD2=0.
      SUMEYD=0.
      SUMEYA=0.
      TIME=0.
      C1=0.
      C2=0.
      COST=0.
      Y1=YASS
      Y2=YDSS
      FLUX=FSS
      TEMP=TSS
      J=0
      NPT=0
      TAU1=TAUAT

```

```

      TAU4=TAUAT
      TAU2=TAUAF
      TAU3=TAUAF
      DUMYAL=0.
      DUMYDL=0.
      COMP1=0.
      COMP2=0.
      PRINT81, ITYPE
81    FORMAT(1H1,10X,'** CASE NUMBER ',I3,' **')
      PRINT95,REFYA,REFYD
95    FORMAT(10X,'SET POINTS',2E20.5//)
      IF(ITYPE.LE.3) GOTO51
      PRINT83,DET
83    FORMAT(10X,' DECOUPLER GAIN DENOMINATOR =',E20.5//)
51    CONTINUE
C    CONTROLLER PAIRED CORRECTLY WITH NO INTERACTION COMPENSATION
C    ABSORBER BVHD CONTROLLED WITH AIL FLOW
C    DIST BVHD CONTROLLED WITH FHTR TEMP
      CYAG=0.859/XKAF*(DEADAF/TAUAF)**(-.977)
      CYDG=.859/XKDT*(DEADDT/TAUDT)**(-.977)
      RTIYA=0.674/TAJAF*(DEADAF/TAUAF)**(-.68)
      RTIYD=.674/TAUDT*(DEADDT/TAUDT)**(-.68)
      IF(ITYPE.LE.5) GOTO50
      CYAG=.81
      RTIYA=15.8822
      CYDG=2.17
      RTIYD=.252
50    PRINT30,CYAG,RTIYA,CYDG,RTIYD
30    FORMAT(1X,'CONTROLLER CONSTANTS'//1X,'FFED FLOW',5X,2F20.4//,1X,
1    'FHTR TEMP',5X,2F20.4//)
      PRINT1
1    FORMAT(5X,'TIME',5X,'Y1',10X,'Y2',10X,'FLOW',10X,'TEMP',10X,'COST'
*//)
      PRINT2,TIME,Y1,Y2,FLOW,TEMP,COST

```

```

2      FORMAT(1X,F8.2,5E12.5)
C      INTEGRATION OF MODEL EQUATIONS
C      MODEL = 1ST ORDER LAG PLUS DEAD TIME
      D811 J=1,1401
      J=J+1
      TIME=TIME+H
      EYA=REFYA-Y1
      EYD=REFYD-Y2
      SUMEYA=SUMEYA+H*EYA
      SUMEYD=SUMEYD+H*EYD
      C1=C1+H*TIME*ABS(EYA)
      C2=C2+H*TIME*ABS(EYD)
      COST=C1+C2
      DF=CYAG*(EYA+RTIYA*SUMEYA)
      DT=CYDG*(EYD+RTIYD*SUMEYD)
      IF(ITYPE.EQ.4) G8T870
      IF(ITYPE.EQ.1) DT=0.
      IF(ITYPE.EQ.2) DF=0.
      G8T871
99      DUMYA=0.3E-07
      DUMYD=0.
      DF=XKDT*DUMYA/DET
      DT=XKDF*DUMYA/DET
      G8T871
97      DUMYA=0.
      DUMYD=-.025
      DF=XKAT*DUMYD/DET
      DT=XKAF*DUMYD/DET
      G9T971
69      DUMYA=.3E-07
      DUMYD=0.
      G8T972
68      DUMYA=0.
      DUMYD=-.025

```

```

68T872
70 CONTINUE
C DECOUPLER EQUATIONS -- USE SQ GAINS ONLY
  IF(ITYPE.EQ.4) 68T999
  IF(ITYPE.EQ.5) 68T997
  IF(ITYPE.EQ.6) 68T969
  IF(ITYPE.EQ.7) 68T968
  DUMYA=DF
  DUMYD=DT
  IF(ITYPE.EQ.8) DUMYD=0.
  IF(ITYPE.EQ.9) DUMYA=0.
72 CONTINUE
  C0MP1=(TAU1*(DUMYA-DUMYAL)+H*DUMYA+TAU2*C0MP1)/(TAU2+H)
  C0MP2=(TAU3*(DUMYD-DUMYDL)+H*DUMYD+TAU4*C0MP2)/(TAU4+H)
  DUMYDL=DUMYD
  DUMYAL=DUMYA
  DT=(XKAF*DUMYD-XKDF*C0MP1)/DET
  DF=(XKDT*DUMYA-XKAT*C0MP2)/DET
71 CONTINUE
  DYA1=DYA1+H*(XKAF*DF-DYA1)/TAUAF
  DYA2=DYA2+H*(XKAT*DT-DYA2)/TAUAT
  DYD1=DYD1+H*(XKDF*DF-DYD1)/TAUDF
  DYD2=DYD2+H*(XKDT*DT-DYD2)/TAUDT
  DYA1D=DDYA1(1)
  NDUM=NYA1-1
  D821 II=1,NDUM
  DDYA1(II)=DDYA1(II+1)
  DDYA1(NYA1)=DYA1
  NDUM=NYA2-1
  DYA2D=DDYA2(1)
  D822 II=1,NDUM
  DDYA2(II)=DDYA2(II+1)
  DDYA2(NYA2)=DYA2
  NDUM=NXD1-1

```



```

23      DDYD1=DDYD1(1)
      DD23 II=1,NDUM
      DDYD1(II)=DDYD1(II+1)
      DDYD1(NYD1)=DYD1
      NDUM=NYD2-1
      DYD2D=DDYD2(1)
      DD24 II=1,NDUM
      DDYD2(II)=DDYD2(II+1)
      DDYD2(NYD2)=DYD2
      DYA=DYA1D+DYA2D
      DYD=DDYD1D+DYD2D
      Y1=YASS+DYA
      Y2=YDSS+DYD
      FLOW=FSST+DF
      TEMP=TSS+DT
      IF(J.NE.10) GOTO11
      PRINT2,TIME,Y1,Y2,FLW,TEMP,COST
      J=0
      NPT=NPT+1
      XPTS(NPT)=TIME
      YPTS(NPT)=Y1
      YYPTS(NPT)=Y2
      ZPTS(NPT)=FLOW
      ZZPTS(NPT)=TEMP
      IF(Y1.LT.Y1MIN) YPTS(NPT)=Y1MIN
      IF(Y1.GT.Y1MAX) YPTS(NPT)=Y1MAX
      IF(Y2.LT.Y2MIN) YYPTS(NPT)=Y2MIN
      IF(Y2.GT.Y2MAX) YYPTS(NPT)=Y2MAX
      IF(FLW.LT.FMIN) ZPTS(NPT)=FMIN
      IF(FLW.GT.FMAX) ZPTS(NPT)=FMAX
      IF(TEMP.LT.TMIN) ZZPTS(NPT)=TMIN
      IF(TEMP.GT.TMAX) ZZPTS(NPT)=TMAX
11      CONTINUE
C      ROUTINE FOR PLOTTING AJT RESULTS

```

```

CALL PLAT(0.,0.,-3)
XPTS(NPT+1)=0.
XPTS(NPT+2)=2.
YPTS(NPT+1)=Y1*IN
YPTS(NPT+2)=.1E-06
YYPTS(NPT+1)=Y2*IN
YYPTS(NPT+2)=.05
ZPTS(NPT+1)=0.
ZZPTS(NPT+1)=0.
ZPTS(NPT+2)=20.
ZZPTS(NPT+2)=300.
CALL AXIS(0.,0.,YAXIS,02,5.,90.,YPTS(NPT+1),YPTS(NPT+2))
CALL AXIS(0.,0.,XAXIS,-4,7.,0.,XPTS(NPT+1),XPTS(NPT+2))
CALL PLAT(7.,0.,3)
CALL PLAT(7.,5.,2)
CALL PLAT(0.,5.,2)
CALL PLAT(0.,0.,3)
CALL FLINE(XPTS,YPTS,NPT,1,0,0)
CALL PLAT(0.,7.,-3)
CALL AXIS(0.,0.,YYAXIS,02,5.,90.,YYPTS(NPT+1),YYPTS(NPT+2))
CALL AXIS(0.,0.,XAXIS,-4,7.,0.,XPTS(NPT+1),XPTS(NPT+2))
CALL PLAT(7.,0.,3)
CALL PLAT(7.,5.,2)
CALL PLAT(0.,5.,2)
CALL PLAT(0.,0.,3)
CALL FLINE(XPTS,YYPTS,NPT,1,0,0)
CALL PLAT(12.,-7.,-3)
IF(ITYPE.LE.7) GO TO 100
CALL AXIS(0.,0.,ZAXIS,1,5.,90.,ZPTS(NPT+1),ZPTS(NPT+2))
CALL AXIS(0.,0.,XAXIS,-4,7.,0.,XPTS(NPT+1),XPTS(NPT+2))
CALL PLAT(7.,0.,3)
CALL PLAT(7.,5.,2)
CALL PLAT(0.,5.,2)
CALL PLAT(0.,0.,3)

```

```

CALL FLINE(XPTS,ZPTS,NPT,1.0,0)
CALL PLBT(0.,7.,-3)
CALL AXIS(0.,0.,ZZAXIS,01,5.,9.,ZPTS(NPT+1),ZPTS(NPT+2))
CALL AXIS(0.,0.,XAXIS,-4,7.,0.,XPTS(NPT+1),XPTS(NPT+2))
CALL PLBT(7.,0.,3)
CALL PLBT(7.,5.,2)
CALL PLBT(0.,5.,2)
CALL PLBT(0.,0.,3)
CALL FLINE(XPTS,ZPTS,NPT,1.0,0)
CALL PLBT(12.,-7.,-3)
100 CONTINUE
CALL PLBT(0.,0.,999)
STOP
END

```

```

C
C PROGRAM AND CONTROL CARDS TO RUN CHESS ON THE SIGMA 5
C INCLUDES DATA FOR SAMPLE PROGRAM IN CHESS MANUAL
C
C IJ98
C IATTEND
C IALL9BT (FILE,GG),(FORMAT,R),(FSIZE,30),(RFSIZE,30)
C IALL9BT (FILE,6V),(FSIZE,312)
C IFORTRANH GR
C COMM6N/SYSAA/ TITLE(20),COMPNT(10),KAMNAM(40)
C COMM6N/SYSA/ KPY( 9,25),KSFH(3.50 ),N3MAX
C COMM6N/SYSB/ KE1(25),NE1MAX,KE2(25),NE2MAX,KE3(25),NE3MAX
C COMM6N/SYSB/KRET,KRET2,KRET3
C COMM6N/SYSC/ LIMIT,LIMIT2,LIMIT3,L89P,L89PS
C COMM6N/SYSD/ KEFLAG(25),KSFLAG(50 ),KTRACE,DERR9R
C COMM6N/EQPA/ EQPAR(25,25),NEMAX,MAXEQP
C COMM6N/EQPB/NECALL(25),NAMF(25)
C COMM6N/STMA/ SEXTSV(13,50 ),SINTSV( 6,50 ),NSMAX,MAXSEX,MAXSIN
C COMM6N/STRMIN/SINUM(4),SIFLAG(4),STVPR(4),STTFMP(4),SIPRES(4),
1SIENTH(4),SIM8LE(4),SIC8MP(10,4),SIKV(10,4)
C COMM6N/STM8UT/SRNUM(4),S8FIAG(4),SAVPR(4),S8TEMP(4),S8PRES(4),
1S8ENTH(4),S8M8LF(4),S8C8MP(10,4),SAKV(10,4)
C COMM6N/C8NTL/NIN,NSUT,N8C8MP,NF,NEN
C COMM6N/M8ZK/BASEA,BASEB,ZCD,AID,LNPHI,LNACT,I NNU
C COMM6N/MHX/SAVEJT(10)
C COMM6N/HXMS/NESAVE(12)
C COMM6N/PHD/ APC(10),ATC(10),AVC(10),AMW(10),ARMEG(10),ADEL(10),
X AVX(10),APH(10),BET(10),SAM(10),DTA(10),EXFLAG
C COMM6N/YMSEQ/INT(17,24),SAVFSW
C DOUBLE PRECISION BASFA(10),BASFB(10),LNNU(10),LNACT(10),LNPHI(10)
C DOUBLE PRECISION ZCD(10)
C REAL ALD(10)
C CONTINUE
15 CALL SEGL9D(2)

```

MAND00010
 MAND00020
 MAND00030
 MAND00040
 MAND00050
 MAND00060
 MAND00070
 MAND00080
 MAND00090
 MAND00100
 MAND00110
 MAND00120
 MAND00130
 MAND00140
 MAND00150
 MAND00160
 MAND00170
 MAND00180
 MAND00190
 MAND00200
 MAND00210
 MAND00220
 MAND00230
 MAND00240
 MAND00250

CALL DRFAD
CALL COMPID
CALL SEGLBD(10)
CALL SEGLBD(18)
CALL INIT
CALL SEGLBD(2)
CALL PTRNT
CALL EQPRT
CALL SEGLBD(10)
CALL SUBSET
CALL SEGLBD(2)
CALL BIPRT
CALL PTEQPF
GOTO 15
END

MAN00260
MAN00270
MAN00280
MAN00290
MAN00300
MAN00310
MAN00320
MAN00330
MAN00340
MAN00350
MAN00360
MAN00370
MAN00380
MAN00390
MAN00400

```

ZER00010
ZER00020
ZER00030
ZER00040
ZER00050
ZER00060
ZER00070
ZER00080
ZER00090
ZER00100
ZER00110

```

```

SUBROUTINE ZER0 (A,N)
  DIMENSION A(N)
  DO 3 I=1,N
    3 A(I) =0.
  RETURN
  ENTRY ZER0X(K,N)
  DIMENSION K(N)
  DO 1 I=1,N
    1 K(I)=0
  RETURN
  END

```

```

IALLBRT (FILE,X1),(FSIZE,0),SAVE
IALLBRT (FILE,X6),(FSIZE,1)
IALLBRT (FILE,X2),(FSIZE,115)
IALLBRT (FILE,X3),(FSIZE,10)
IALLBRT (FILE,X4),(FSIZE,10)
IALLBRT (FILE,X5),(FSIZE,10)
ILOAD (TEMP,205),(UDCB,2),(MAP,PROGRAM),(F0RE,1600)
:R00T (FILE,B1,G0,2)
:SEG (LINK,2,0NT0,0),(FILE,FP,CSREAD,2)
:SEG (LINK,10,0NT0,0),(FILE,FP,CSKHZT,5)
:SEG (LINK,18,0NT0,10),(FILE,SP,CSSCAN,1)
:SEG (LINK,27,0NT0,10),(FILE,SP,CSHXFR,2)
:SEG (LINK,16,0NT0,10),(FILE,FP,CSPRA1,3)
:ASSIGN (F:5,SI)
:ASSIGN (F:6,L0)
IR0V
CLEAN,NAMELIST
CRACKING AND ALKYLATION PROCESS - MATERIAL AND ENRGY BALANCES
NAME=3,8,11,5,5,5,7,3,2,8,11,5,6,7,2,7,2,3,8,0,0,0,0,0,0,
KPM1=1,3,1,25,-2,0,0,0,0,
KPM2=2,8,2,-3,0,0,0,0,0,
KPM3=3,11,3,-4,0,0,0,0,0,
KPM4=4,5,4,-5,0,0,0,0,0,
KPM5=5,5,5,-6,0,0,0,0,0,
KPM6=6,5,6,-7,0,0,0,0,0,
KPM7=7,7,7,-8,0,0,0,0,0,
KPM8=8,3,8,23,-9,0,0,0,0,
KPM9=9,2,9,-10,-11,0,0,0,0,
KPM10=10,8,11,-12,0,0,0,0,0,
KPM11=11,11,12,-13,0,0,0,0,0,
KPM12=12,5,13,-14,0,0,0,0,0,
KPM13=13,6,14,-15,0,0,0,0,0,
KPM14=14,7,15,-16,0,0,0,0,0,
KPM15=15,2,16,-18,-17,0,0,0,0,

```



```

EQP17=17,,0,,0,,0,,1,,1,,0,,0,,0,,0,,0,,0,,0,,
0,,0,,0,,0,,0,,0,,0,,0,,0,,0,,
EQP19=19,,1,,500000,,1120,,0,,0,,0,,0,,0,,0,,0,,0,,0,,
0,,0,,0,,0,,0,,0,,0,,0,,0,,
SEX1=1,,0,,16,,0,,0,,10,,5,,1,,0,,0,,0,,0,,0,,
SEX21=21,,0,,7,5,0,,0,,0,,0,,7,5,0,,0,,0,,0,,0,,
SNAME=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,23,25,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
SNAME=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,23,25,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
SIN1=1,,1,,1,,530,,98,,0,,
SIN10=10,,2,,1,,0,,0,,0,,
SIN17=17,,2,,0,,0,,0,,0,,
SIN21=21,,1,,1,,460,,98,,0,,
KE2=1,2,3,4,5,6,7,18,19,8,9,10,11,12,13,14,15,16,17,0,0,0,0,0,
L00PS=10, DERR0R=.001,
IE00
IFIN

```

VITA

William Harold Pusch, Jr. was born in New Orleans, Louisiana on July 1, 1945. He obtained his elementary and secondary education in the Jefferson Parish school system and graduated from East Jefferson High School in May 1963.

He attended Louisiana State University and received his Bachelor of Science Degree in Chemical Engineering in May 1967. He was awarded a Master of Science Degree in Chemical Engineering in January 1969.

At present he is a candidate for the degree of Doctor of Philosophy in Chemical Engineering at Louisiana State University.

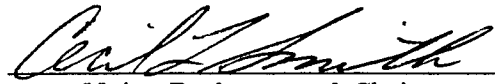
EXAMINATION AND THESIS REPORT

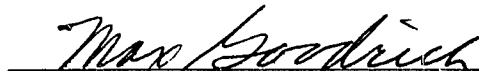
Candidate: William Harold Pusch, Jr.

Major Field: Chemical Engineering


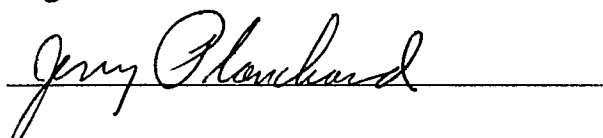
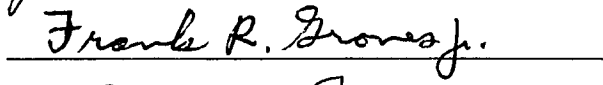

Title of Thesis: Practical Applications of Advanced Control Theory

Approved:


Major Professor and Chairman


Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

November 19, 1971